Aplikasi Pengamanan Data dengan Teknik Algoritma Kriptografi AES dan Fungsi Hash SHA-1 Berbasis Desktop

Ratno Prasetyo
Magister Ilmu Komputer
Universitas Budi Luhur, Jakarta, 12260
Telp : (021) 5853753 ext 253, Fax : (021)
er.prast@gmail.com

Asep Suryana
Magister Ilmu Komputer
Universitas Budi Luhur, Jakarta, 12260
Telp : (021) 5853753 ext 253, Fax : (021)
asep.suryanammi@gmail.com

Abstrak- **Dampak positif kemajuan teknologi informasi dapat membantu menyelesaikan pekerjan dan pertukaran data dan informasi dengan cepat, akurat, dan efisien. Namun disisi lain, terdapat dampak negative berupa penyadapan yang memngakibatkan suatu data dan informasi diambil atau dimiliki oleh pihak yang tidak memiliki otoritas atau hak akses untuk merubahnya. Agar data tersebut aman dari pihak-pihak yang tidak diotorisasi maka dibuatlah Aplikasi Pengamanan Data dengan teknik kriptografi. Teknik yang digunakan dalam tulisan ini menggunakan Algoritma kriptografi AES (Advanced Encryption Standard) serta fungsi Hash SHA-1. Aplikasi pengamanan data berbasis desktop ini dibuat menggunakan bahasa pemrograman Visual Basic.NET. Aplikasi ini mempunyai 2 (dua) fasilitas antara lain: enkrip berkas dan dekrip berkas.**

Kata kunci-- Kriptografi, AES, SHA-1, Enkripsi, Dekripsi

I. PENDAHULUAN

Kriptografi merupakan studi matematika yang mempunyai hubungan dengan aspek keamanan informasi seperti integritas data, keaslian entitas dan keaslian data. Kriptografi menggunakan berbagai macam teknik dalam upaya untuk mengamankan data. Pengiriman data dan penyimpanan data melalui media elektronik memerlukan suatu proses yang dapat menjamin keamanan dan keutuhan dari data yang dikirimkan tersebut. Data tersebut harus tetap rahasia selama pengiriman dan harus tetap utuh pada saat penerimaan di tujuan. Untuk memenuhi hal tersebut, dilakukan proses penyandian (enkripsi dan dekripsi) terhadap data yang akan dikirimkan.

Enkripsi dilakukan pada saat pengiriman dengan cara mengubah data asli menjadi data rahasia, sedangkan dekripsi dilakukan pada saat penerimaan dengan cara mengubah data rahasia menjadi data asli. Jadi data yang dikirimkan selama proses pengiriman adalah data rahasia, sehingga data asli tidak dapat diketahui oleh pihak yang tidak berkepentingan. Data asli hanya dapat diketahui oleh penerima dengan menggunakan kunci rahasia.

Disini enkripsi dapat diartikan sebagai kode atau *cipher*. Sebuah *system* pengkodean menggunakan suatu tabel atau kamus yang telah didefinisikan untuk kata dari informasi atau yang merupakan bagian dari pesan, data, atau informasi yang di kirim. Sebuah cipher menggunakan suatu algoritma yang dapat mengkodekan semua aliran data *(stream)* bit dari suatu pesan asli *(plaintext)* menjadi *cryptogram* yang tidak di mengerti. Karena *system cipher* merupakan suatu sistem yang telah siap untuk di outomasi, maka teknik ini digunakan dalam sistem keamanan jaringan komputer.

National Institute of Standard and Technology (NIST) untuk pertama kalinya mengumumkan suatu algoritma standar penyandian data yang telah dijadikan standar sejak tahun 1977 adalah *Data Encryption Standard* (*DES*). Kekuatan *DES* ini terletak pada panjang kuncinya yaitu 56-bit. Untuk menanggapi keinginan agar mengganti algoritma *DES* sebagai standar. Perkembangan kecepatan perangkat keras dan meluasnya penggunaan jaringan komputer terdistribusi mengakibatkan penggunaan *DES*, dalam beberapa hal, terbukti sudah tidak aman dan tidak mencukupi lagi terutama dalam hal yang pengiriman data melalui jaringan internet. Perangkat keras khusus yang bertujuan untuk menentukan kunci 56-bit *DES* hanya dalam waktu beberapa jam sudah dapat dibangun. Beberapa pertimbangan tersebut telah manandakan bahwa diperlukan sebuah standar algoritma baru dan kunci yang lebih panjang. *Triple-DES* muncul sebagai *alternative* solusi untuk masalah-masalah yang membutuhkan keamanan data tingkat tinggi seperti perbankan, tetapi ia terlalu lambat pada beberapa penggunaan enkripsi.

Pada tahun 1997, *the U.S. National Institue of Standards and Technology (NIST)* mengumumkan bahwa sudah saatnya untuk pembuatan standard

algoritma penyandian baru yang kelak diberi nama *Advanced Encryption Standard (AES)*. Algoritma *AES* ini dibuat dengan tujuan untuk menggantikan algoritma *DES* & *Triple-DES* yang telah lama digunakan dalam menyandikan data elektronik. Setelah melalui beberapa tahap seleksi, algoritma *Rijndael* ditetapkan sebagai algoritma kriptografi *AES* pada tahun 2000.

Algoritma *AES* merupakan algoritma kriptografi simetrik yang beroperasi dalam mode penyandi blok *(block cipher)* yang memproses blok data 128-bit dengan panjang kunci 128-bit *(AES- 128)*, 192-bit *(AES-192)*, atau 256-bit *(AES-256)*.

A. Batasan Masalah

Batasan-batasan masalah dalam tulisan ilmiah ini adalah sebagai berikut :

- Sebelum data disembunyikan terlebih dahulu dilakukan penyandian dengan *password* yang dikonversi menjadi *byte* menggunakan *SHA-1* kemudian *file* dienkripsi menggunakan algoritma *AES (Advance Encryption Standard)* 128 bit.
- Proses otentikasi *password* menggunakan fungsi *hash SHA-1*.
- Data yang dienkripsi merupakan *file text* atau *document, file image*, *file zip*, *file audio*, dan *file video*.
- Program dibuat dengan menggunakan bahasa pemrograman *Visual Basic .NET*.

II. TINJAUAN STUDI

Kriptografi *(Cryptography)* berasal dari bahasa yunani, *cyptos* artinya *secret* atau rahasia sedangkan *graphein* berarti *writing* atau tulisan. Sehingga kriptografi berarti *secret writing* atau tulisan rahasia. Kriptografi adalah ilmu yang bersandarkan pada teknik matematika untuk berurusan dengan keamanan informasi seperti kerahasiaan, keutuhan data, dan otentikasi entitas. [2]. Selain itu juga Kriptografi dapat diartikan sebagai ilmu untuk menjaga kerahasiaan informasi dengan metode dan teknik matematika yang mencakup *confidentiality, integrity, authentication* dan *non-repudiation* [1].

A. Kriptografi Sistem Simetris

Sistem simetris adalah sistem kriptografi yang menggunakan kunci yang sama untuk proses enkripsi dan dekripsi [5]. Sistem ini sering juga disebut dengan algoritma kunci tunggal atau algoritma satu kunci. Bila E adalah fungsi enkripsi *(encryption)*, D adalah fungsi dekripsi *(decryption)*, K adalah kunci rahasia *(key)*, sedangkan M adalah pesan orisinil yang akan dikirimkan *(message)* dan C adalah pesan sandinya *(cipher)*, maka sistem simetris dapat diformulasikan sebagai berikut :

$$E_k(M)=C \text{ dan } D_k(C)=M \quad .... (2.1)$$

Dalam aplikasinya antara pengirim dan penerima harus ada persetujuan atau sinkronisasi kunci agar saling berkomunikasi. Jadi, keamanan algoritma sistem simetris terletak pada kunci. Siapapun yang memperoleh kunci, akan dapat membuka pesan yang dikomunikasikan. Karena itu selama proses komunikasi bersifat rahasia, maka kunci harus tetap dirahasiakan. Sistem Kripto Simetris yang menyediakan keamanan secara praktis terbagi dua kategori yaitu *Stream Cipher* dan *Block Cipher*.

B. Algoritma Kriptografi AES

Input dan *output* dari algoritma *AES* terdiri dari urutan data sebesar 128 *bit*. Urutan data yang sudah terbentuk dalam satu kelompok 128 *bit* tersebut disebut juga sebagai blok data atau *plaintext* yang nantinya akan dienkripsi menjadi *ciphertext*. *Cipher key* dari *AES* terdiri dari *key* dengan panjang 128 *bit*, 192 *bit*, atau 256 *bit*. Perbedaan panjang kunci akan mempengaruhi jumlah *round* yang akan diimplementasikan pada algoritma *AES* ini. Berikut ini adalah tabel yang memperlihatkan jumlah putaran *(round)* yang harus diimplementasikan pada masing-masing panjang kunci dengan catatan 1 *word* = 32 bit.

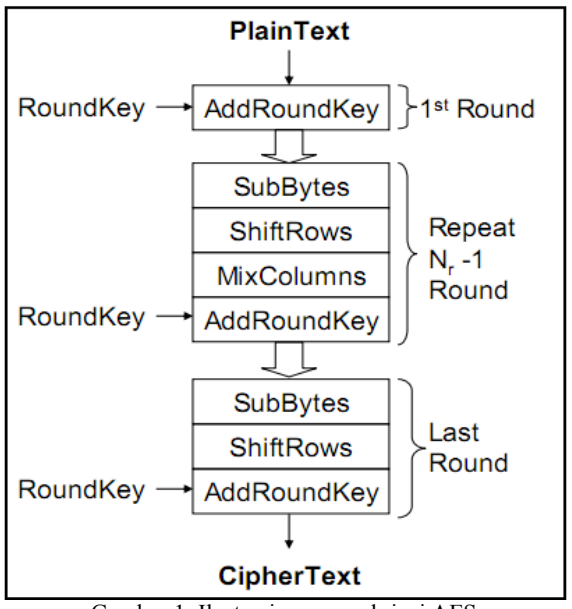
Tabel 1. Perbandingan Jumlah Putaran dan Kunci [3]

	Jumlah Kunci (Nk)	Jumlah Putaran (Nr)
AES-128	4	10
AES-192	6	12
AES-256	8	14

Dalam hal ini data *AES* mempunyai 5 ukuran unit data yaitu : *bit* merupakan satuan data terkecil, yaitu nilai digit sistem biner, *byte* berukuran 8 *bit*, *word* berukuran 4 *byte* (32 *bit*), *block* berukuran 16 *byte* (128 *bit*) dan *state* merupakan *block* yang ditata sebagai matrik *byte* berukuran 4×4.

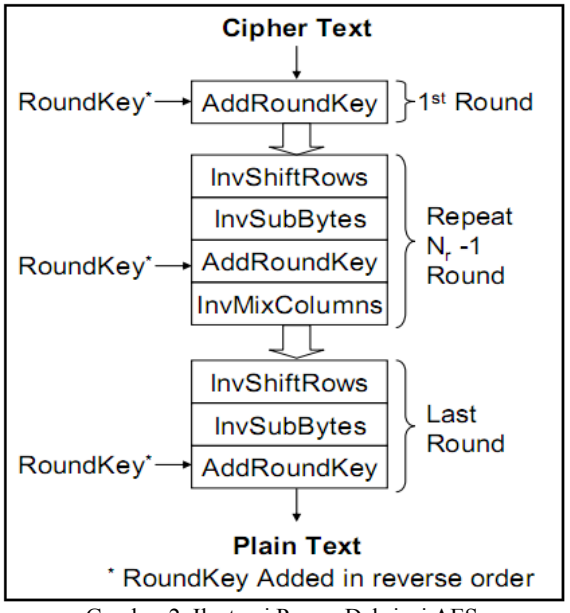
C. Algoritma AES

Proses enkripsi algoritma *AES* terdiri dari 4 jenis transformasi *bytes*, yaitu *SubBytes, ShiftRows, Mixcolumns*, dan *AddRoundKey*. Pada awal proses enkripsi, *input* yang telah disalin ke dalam *state* akan mengalami transformasi *byte AddRoundKey*. Setelah itu, *state* akan mengalami transformasi *SubBytes*, *ShiftRows*, *MixColumns*, dan *AddRoundKey* secara berulang-ulang sebanyak Nr. Proses ini dalam algoritma *AES* disebut sebagai *round function*. *Round* yang terakhir agak berbeda dengan *round-round* sebelumnya dimana pada *round* terakhir, *state* tidak mengalami transformasi *MixColumns* [8]. Ilustrasi proses enkripsi *AES* dapat digambarkan seperti pada Gambar 1.



Gambar 1. Ilustrasi proses enkripsi AES

Transformasi *cipher* dapat dibalikkan dan diimplementasikan dalam arah yang berlawanan untuk menghasilkan *inverse cipher* yang mudah dipahami untuk algoritma *AES*. Transformasi *byte* yang digunakan pada *invers cipher* adalah *InvShiftRows*, *InvSubBytes*, *InvMixColumns*, dan *AddRoundKey* [3]. Algoritma dekripsi dapat dilihat pada Gambar 2.



Gambar 2. Ilustrasi Proses Dekripsi AES

D. Hash SHA-1

Fungsi *hash* merupakan sebuah algoritma yang mengubah teks atau pesan menjadi sederetan karakter acak yang memiliki jumlah karakter yang sama. *Hash* juga termasuk salah satu bentuk teknik kriptografi dan dikategorikan sebagai kriptografi tanpa kunci *(unkeyed cryptosystem)*. Hal yang mendasar yang menjadi perbedaan dari fungsi *hash* adalah pesan yang telah acak tidak dapat diubah kembali menjadi pesan terang.

Menurut Kaufman et. al fungsi *hash* dapat digunakan sebagai:

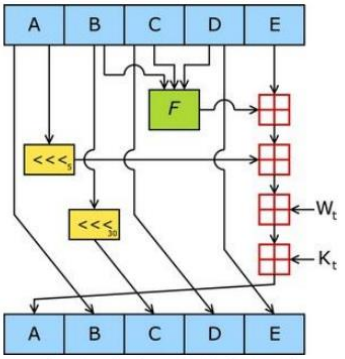
- Untuk menyimpan *password*.
- Sebagai *Message Integrity*.
- Sebagai *Message Fingerprint.*

Fungsi *Hash* digunakan untuk menjamin data atau pesan yang dikirim dan tidak mengalami modifikasi, pemalsuan atau injeksi selama transmisi (*Message Integrity*). Suatu fungsi *hash* akan memetakan *bit-bit string* dengan panjang sembarang ke sebuah *string* dengan panjang tertentu misal n. Proses pemetaan suatu *input string output* tersebut disebut dengan proses *hashing*. *Output* dari fungsi *hash* disebut dengan nilai *hash*, kode *hash* atau hasil *hash*.

Cara kerja kriptografi algoritma *SHA-1* adalah menerima *input* berupa pesan dengan ukuran sembarang dan menghasilkan *message digest* yang memiliki panjang 160 bit. Langkah-langkah pembuatan *message digest* dengan algoritma *SHA-1* adalah sebagai berikut :

- *Input* Pesan yang akan di *hash SHA-1*.
- Ubah pesan menjadi deretan biner
- Penambahan Bit-bit pengganjal, yaitu dengan menambahkan pesan dengan sejumlah bit pengganjal sedemikian sehingga panjang pesan (dalam satuan bit) *kongruen* dengan 448 mod 512.
- Penambahan nilai panjang pesan semula, yaitu pesan ditambah lagi dengan 64 bit yang representasi *biner* dari panjang pesan asli.
- Inisialisasi Nilai *Hash*, pada algoritma SHA-1 nilai *hash*, H(0) terdiri dari 5 *words* dengan besar 32 bit dalam notasi *hexadecimal*.
- *Output* nilai *hash* adalah nilai terakhir dari *buffer*.

Berdasarkan tahapan yang ada pada Fungsi *Hash SHA-1*, maka skema Fungsi *Hash SHA-1* dapat dilihat pada gambar berikut ini:



Gambar 3. Skema Fungsi Hash SHA-1 [4]

III. ANALISA MASALAH, DAN RANCANGAN PROGRAM

A. *Analisa Masalah*

Kemudahan dalam pertukaran data dan informasi melalui jaringan terbuka dan luas seperti internet, berbanding lurus dengan bahaya yang mengancam dari pihak luar terhadap keamanan atau keutuhan data atau informasi tersebut. Pada *system* komunikasi terbuka, peluang pihak lain untuk memperoleh data atau informasi yang dikomunikasikan lebih besar dan penyadap yang berrhasil memperoleh informasi tersebut dapat langsung memahami isinya. Sejalan dengan ancaman terhadap keamanan data, telah berkembang beberapa teknik atau cara pengamanan data.

Pengamanan data dapat dilakukan dengan mengacak data sebenernya menjadi informasi yang tidak dapat terbaca atau disebut kriptografi. Pengamanan data ini dilakukan dengan tujuan untuk menyampaikan data penting atau rahasia kepada penerima yang berhak.

B. *Strategi Pemecahan Masalah*

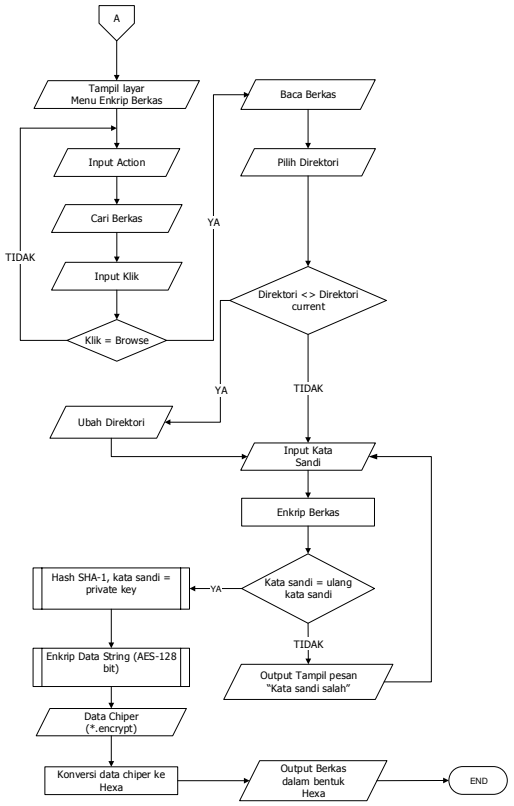
Dalam penyusunan tulisan ilmiah ini, penulis merancang aplikasi pengamanan data berbasis *desktop* dengan mengimplementasikan fungsi *hash SHA-1 (Secure Hash Algorithm-1)* dan kriptografi menggunakan Algoritma *AES*. Dimana kebutuhan sistem yang akan dibangun pada aplikasi ini adalah sebagai berikut :

- Proses pengamanan data menggunakan aplikasi yang berbasis *desktop*.
- Algoritma enkripsi yang digunakan untuk mengamankan data harus sesuai dengan teknik kriptografi untuk tetap menjaga keutuhan berkas/*file* tersebut ketika di dekripsi.
- Keamanan data harus terjamin dengan menggunakan algoritma enkripsi yang kuat, dalam hal ini penulis menggunakan Algoritma Kriptografi *AES (Advanced Encryption Standard) 128 bit*.

Teknik pengamanan data ini diharapakan lebih memperkuat pengamanan data yang dapat berupa berkas/*file text* atau *document*, *image*, *zip, audio* dan *video* yang akan dikomunikasikan melalui saluran *internet*.

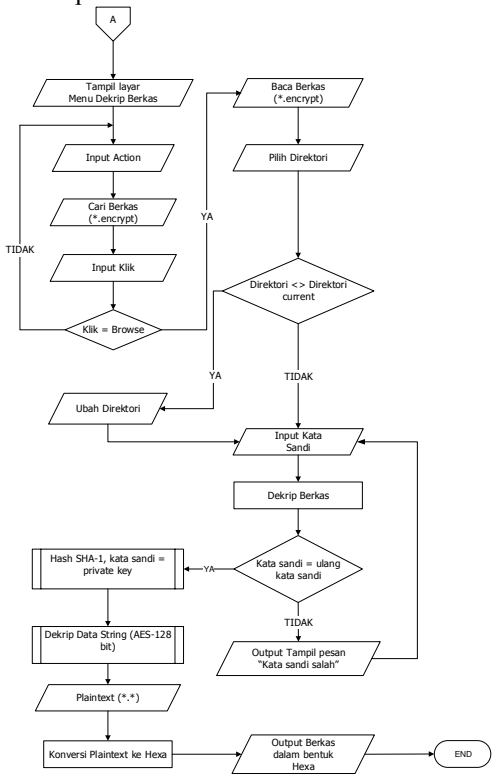
C. *Diagram Alir (Flowchart)*

Flowchart merupakan diagram yang menunjukkan bagaimana cara kerja aplikasi. Sesuai pembahasan diatas, maka diagram alir *(flowchart)* pengguna aplikasi yang terdiri dari proses enkripsi, dan proses dekripsi. Diagram alir proses enkripsi menjelaskan perubahan data atau berkas/*file* menjadi suatu berkas/*file* *.encrypt. Diagram alir proses enkripsi dapat dijelaskan pada Gambar 4.



Gambar 4. Flowchart Enkrip Berkas

Diagram alir proses dekripsi menjelaskan perubahan data atau berkas/*file* *.encrypt menjadi berkas/*file* asli. Diagram alir proses dekripsi dapat dijelaskan pada Gambar 5.



Gambar 5. Flowchart Dekrip Berkas

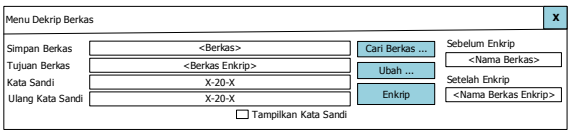
IV. IMPLEMENTASI DAN ANALISA HASIL UJI COBA PROGRAM

A. *Implementasi Program*

Implementasi program akan dibagi ke dalam dua bagian utama yaitu enkripsi/dekripsi dan otentikasi *password*/kata sandi. Program akan memanggil fungsi yang ada pada baris program proses enkripsi dan dekripsi menggunakan Algoritma *AES 128-bit* dengan menggunakan bahasa pemrograman *Visual Basic .NET*.

B. *Tampilan Menu Enkrip Berkas*

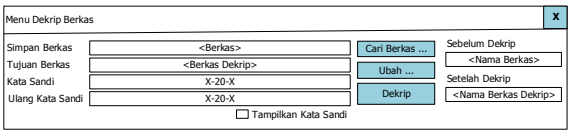
Pengguna dapat melakukan proses enkrip berkas/*file* dengan kata sandi maupun tanpa kata sandi.



Gambar 6. Tampilan Menu Enkrip Berkas

C. *Tampilan Menu Dekrip Berkas*

Pengguna dapat melakukan proses dekrip berkas/*file* dengan kata sandi maupun tanpa kata sandi.



Gambar 7. Tampilan Menu Dekrip Berkas

D. *Hasil Uji Coba Program*

Berdasarkan hasil pengujian program diatas, dapat dijelaskan beberapa hal sebagai berikut :

- Program aplikasi dapat mengenkripsi semua jenis *file (document, image, audio, zip* dan *video)*.
- Ukuran *file* asli lebih kecil dari ukuran *file* yang sudah dienkrip.
- Selisih ukuran *file* sangat tergantung jenis *file* , hal ini disebabkan setiap jenis *file* memiliki karakteristik kerapatan *bit* yang berbeda.
- *File* hasil proses dekrip tidak mengalami kerusakan dan dapat dibaca kembali.
- Kecepatan proses enkrip dan dekrip sangat tergantung pada spesifikasi *hardware* dan *software* pengguna aplikasi.

V. PENUTUP

Dari hasil pengujian dan analisis dapat disimpulkan bahwa:

- Aplikasi ini dapat digunakan sebagai media pengamanan data dengan cara kriptografi untuk semua *file (document, image, zip, audio* dan *video)*.
- Algoritma AES dapat diimplementasikan atau diterapkan dalam teknik enkripsi dan dekripsi pada pemograman berbasis *desktop.*
- Penggunaan Algoritma *AES* menjadikan aplikasi ini dapat memenuhi kekuatan secara kriptografis.
- Penggunaan fungsi *Hash SHA-1* dapat memperkuat pengamanan *password* melalui proses otentikasi.

DAFTAR PUSTAKA

[1] Menezes, J.A., Oorschot, C.P., & Vanstone, A.S. 1997, *Handbook of Applied Crypthography*. USA: CRC Press LLC.

[2] Sadikin, Rifki. 2012. Kriptografi Untuk Keamanan Jaringan dan Implementasinya dalam Bahasa java. Yogyakarta: Penerbit Andi.

[3] Adiwidya, B.M.D. September 2008, "Algoritma AES (Advanced Encryption Standard) dan Penggunaannya dalam Penyandian Pengompresian Data". Makalah 2008, <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2008/2009/Makalah2008/Makalah0809-090.pdf>.

[4] Angga, Christian, 2011, Analisis Cara Kerja Fungsi Hash Yang Ada, dilihat 30 Oktober 2014, <http://informatika.stei.itb.ac.id/~rinaldi.munir/TA/Makalah_TA%20Christian%20Angga.pdf>

[5] Dwi Prasetia, Heri, Juni 2014, "Pengamanan Data Dengan Teknik Steganografi LSB dan Algoritma Kriptografi AES dan Fungsi *Hash SHA-1* Berbasis *Web*". Tangerang: Universitas Budi Luhur.