ISBN 978-623-93343-2-1
Seminar Nasional Informatika, Sistem Informasi dan Keamanan Siber
(SEINASI-KESI)
Jakarta-Indonesia, 19-20 November 2020

PENGAMANAN DATA FILE TEKS MENGGUNAKAN ALGORITMA AES DAN METODE LSB PADA VIDIO AVI

Ifan Alriansyah[1], Henki Bayu Seta[2], Iin Ernawati[3].

[123]Fakultas Ilmu Komputer
Universitas Pembangunan Nasional Veteran Jakarta
fanalriansyah@gmail.com[1], henkiseta@upnvj.ac.id[2], iinerti@gmail.com[3].
Jl. Rs. Fatmawati, Pondok Labu, Jakarta Selatan, DKI Jakarta, 12450, Indonesia

Abstrak

Banyak sekali kejahatan pencurian data penting yang terjadi di dunia maya. Hal ini terjadi dikarenakan lemahnya keamanan dalam proses pengirimin data. Akibatnya data yang dikirim dapat dicuri atau dimodifikasi oleh pihak luar yang tidak bertanggung jawab. Berawal dari masalah tersebut dibutuhkan suatu alat yang mampu memberikan keamanan tambahan kepada pengguna yang mengirimkan datanya. Berdasarkan hal tersebut tujuan penelitian ini dilakukan guna mencegah terjadinya pencurian data oleh pihak luar. Dengan menggunakan *Advanced Encryption Standard* untuk mengenkripsi pesan serta metode *Significant Bit Insertion* untuk menyisipkan hasil dari enkripsi kedalam Vidio agar tidak menimbulkan kecurigaan yang diprogram menggunakan *Matlab* guna membantu proses enkripsi. Dalam penerapan yang dilakukan penulis yaitu tahapan pengujian sistem, algoritma *AES* dapat menekripsi dan mendekripsi data file text serta metode *LSB* dapat menyisipkan dan mengambil file text yang terenkripsi dari Vidio *AVI*.

Kata kunci: Pencurian, Enkripsi, *AES, LSB, MATLAB.*

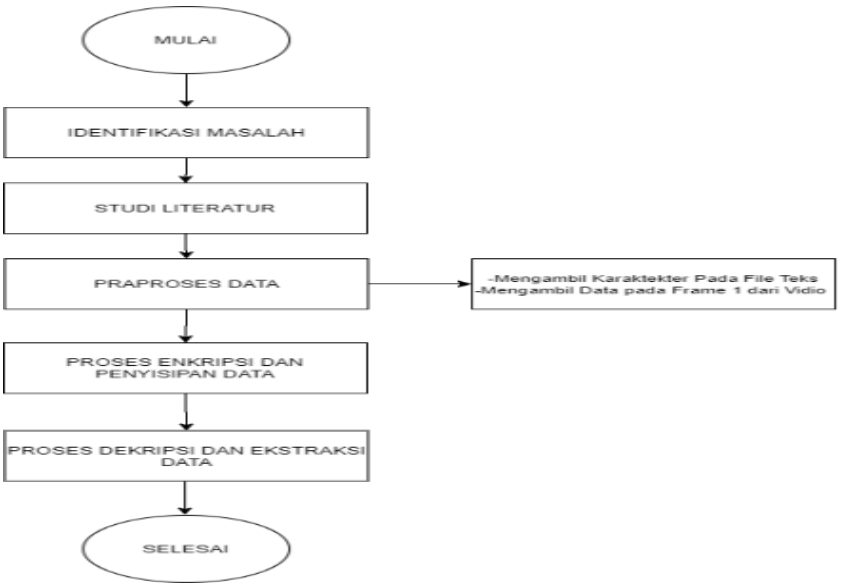
1 PENDAHULUAN

Salah satu teknik keamanan yang sering digunakan pada saat ini adalah kriptografi. Dimana kriptografi tersebut berfungsi sebagai penyembunyian pesan dengan cara mengubah data yang asli menjadi data acak menggunakan kata kunci yang sudah ditentukan, sehingga data yang dikirim pengirim kepada penerima tidak dapat dibaca oleh pihak yang tidak bertanggung jawab. Tetapi dengan menjadikan pesan menjadi kode-kode aneh yang telah di enkripsi maka akan timbul kecurigaan bagi orang yang membacanya sehingga menimbulkan rasa menasaran dan akan berusaha untuk mengetahui kode-kode yang aneh tersebut. Teknik lain selain menggunakan kriptografi adalah dengan steganografi. Teknik ini mengurangai kecurigaan dibandingkan dengan kriptografi dikarenakan pesan penting tidak diubah menjadi kode aneh melainkan disisipkan melalui media lain.

2 METODOLOGI PENELITIAN

2.1 Kerangka Berfikir

Dalam penelitian ini penulis akan melakukan tahapan kerja untuk mencapai tujuan dalam penelitian yang disajikan dalam bentuk *flowchart* pada Gambar 1 dihalaman selanjutnya.



Gambar 1. Kerangka Pikir

2.2 Praproses Data

Pada tahap ini merupakan tahap yang penting agar proses perhitungan pengenkripsian dan penyisipan data dapat berlangsung. Data *file* teks yang diinput akan diambil karakternya saja tidak termasuk gambar, tabel, dan lain-lain hanya teksnya saja. Ketika karakter sudah diambil, karekter akan di konversi ke bentuk hexadecimal yang mana merupakan syarat perhitungan *AES*.

Lalu pada file Vidio yang akan disisipkan diambil data dari *frame* pertama dan membaca layer *RGB* yang masih berbentuk decimal. Lalu diubah ke bentuk biner agar bisa disisipkan pesan yang berbentuk *hexadecimal* yang dikonversi juga ke bentuk biner.

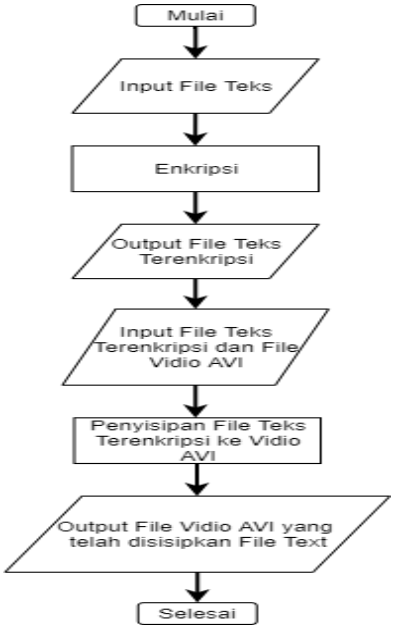
2. Proses Enkripsi dan Penyisipan Data

Tahapan proses enkripsi *AES* yaitu: *AddRoundKey*, *SubBytes*, *ShiftRows*, *MixColumns*. *AddRoundkey* adalah proses mengkombinasikan *state array* dan *roundkey* dengan hubungan *XOR*. Selanjutnya, *SubBytes* adalah proses menukar isi *byte* dengan menggunakan tabel *S-BOX*. *ShiftRows* adalah proses pergeseran blok tiap baris pada *state array*. Terakhir, *MixColumns* yaitu proses mengakalikan blok data di masing masing *state array* dengan Persamaan 1.

$$A(x) = \{03\}x2 + \{01\}x2 + \{01\}x + \{02\} \quad (1)$$

Tahapan tersebut diulang sebanyak 10 kali. Karena pada penelitian kali ini menggunakan *AES*-128. Pada tahap ini file teks dan file vidio yang sudah di praproses sebelumnya akan dilakukan enkripsi dan penyisipan file enkripsi di dalam vidio. Tahapan ini ditampilkan pada Gambar 2 dihalaman selanjutnya.

Gambar 2 merupakan *flowchart* enkripsi dan penyisipan data. Mulai dari input *file* teks yang ingin di enkripsi. Lalu input *file* teks yang sudah di enkripsi beserta input vidio yang menjadi tempat menyisipkan hasil enkripsi berformat *.AVI*. Lalu hasil dari *output* tersebut adalah *file* vidio berformat *.avi* yang berisi data *file* teks yang sudah terenkripsi.

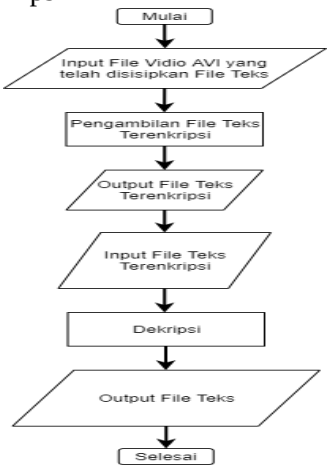


Gambar 2. *Flowchart* Enkripsi dan Penyisipan Data

2.3 Proses Dekripsi dan Ekstrasi Data

Tahapan proses dekripsi *AES* yaitu: *AddRoundKey*, *InvShiftRows*, *InvSubBytes*, *InvMixColumns*. Pada tahap pertama yaitu *AddRoundKey* hasil dari ekstrasi data vidio akan di *XOR* dengan *RoundKey*, tetapi di mulai dari *RoundKey* terakhir yaitu *RoundKey* 10. Selanjutnya *InvShiftRows*, prosesnya sama hanya saja pergeseran yang terjadi tiap barisnya berbeda dengan enkripsi. *InvSubBytes* adalah proses menukar isi *byte* dengan menggunakan tabel *INVERSE* S-*BOX*. Tahapan yang terakhir adalah *InvMixColumns* yaitu mengkalikan blok data di masing-masing *state array* dengan matriks polynomial.

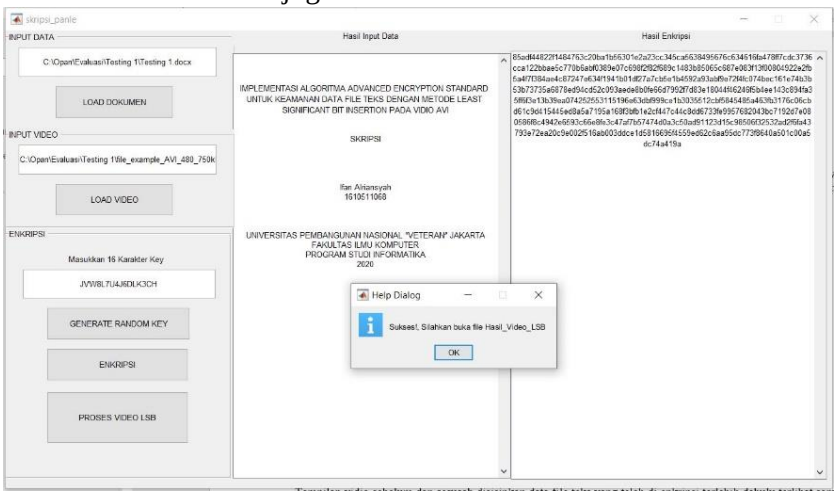
Pada tahap ini vidio yang telah disisipkan *file* enkripsi akan di ekstrak dari vidio supaya mendapatkan *file* enkripsi tersebut. Setelah didapatkan baru di dekripsi supaya *file* teks tersebut dapat dibaca seperti awal. Gambar 3 yang di tampilkan di bawah, merupakan *flowchart* dekripsi dan ekstrasi data. Mulai dari input *file* vidio *.AVI* yang telah disisipkan file teks yang di enkripsi untuk di ekstrak. Setelah di ekstrasi selesai didapatkan *file* teks yang di enkripsi lalu dilakukan proses dekripsi. Lalu hasil dari *output* proses dekripsi adalah *file* data teks awal sebelum di lakukan enkripsi



Gambar 3. *Flowchart* Dekripsi dan Ekstrasi Data

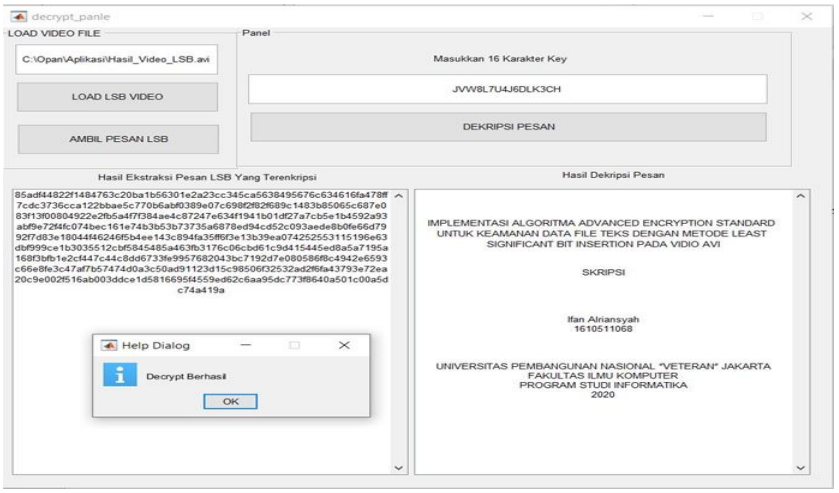
3 Hasil Dan Pembahasan

Pada proses enkripsi pesan text dari *file* .docx berhasil dilakukan penyandian dengan menggunakan *AES*-128 yaitu memakai panjang kunci sebanyak 16 karakter. Kemudian proses penyisipan text ke dalam vidio *.avi* juga berhasil dilakukan.



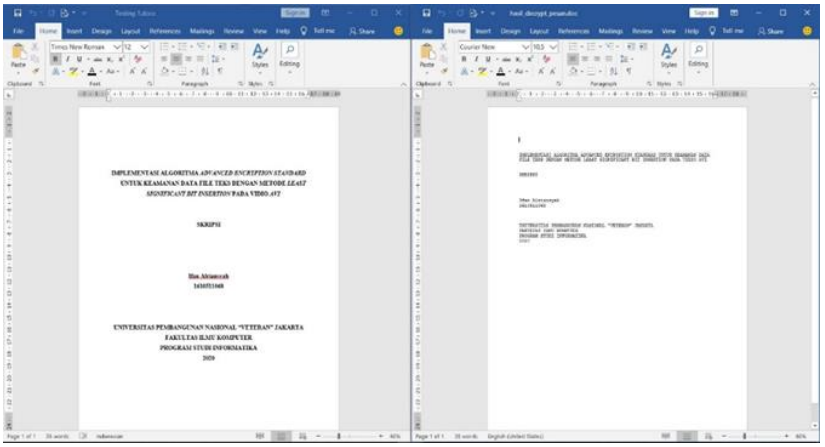
Gambar 1. Hasil Enkripsi dan Penyisipan Data

Selanjutnya proses ekstrasi pesan dari dalam vidio. Pada tahap ini proses ektrasi pesan dari dalam vidio berhasil dilakukan serta pesan tersebut dapat di dekripsi kembali. Proses dekripsi pesan menggunakan kunci yang sama pada proses enkripsi yaitu sebanyak 16 karakter. Hasil ekstrasi dan dekripsi pesan ditampilkan pada Gambar 5 dihalaman selanjutnya.



Gambar. 2. Hasil Ekstrasi dan Dekripsi Pesan

Tampilan file word setelah di lakukan proses enkripsi dan dekripsi serta di sisipkan ke vidio lalu diambil isinya teksnya sama tetapi format pengaturan paragraph, ukuran *font* dan lain-lain tidak. Tampilan file "*Testing 1.docx"* dan "hasil_*decrypt*_pesan.*doc"* ditampilkan pada Gambar 6 dihalaman selanjutnya.



Gambar 3. Tampilan *File Testing* 1 Awal dan Akhir

a) Tampilan Vidio

Tampilan vidio sebelum dan sesusah disisipkan data *file* teks yang telah di enkripsi terlebih dahulu terlihat sama oleh mata. Tetapi ukuran *file* vidio menjadi lebih besar. Tampilan "*file_example_AVI_480_750kB.avi"*ditampilkan pada Gambar 7 dibawah ini.

Gambar 4. Tampilan *File* Vidio Awal

Untuk hasil vidio yang telah disisipkan data file "*Testing 1.docx"* yang telah di enkripsi terlebih dahulu ditampilkan pada Gambar 8 dibawah ini.

Gambar. 5. Tampilan Hasil Vidio *LSB*

Lalu hasil size dari file vidio dan text akan di tambilkan pada *Table* 1.

Table 1: Nama dan Ukuran File

No	Nama File	Size
1	Testing 1.docx	13.6 kb
2	Hasil_decrypt_pesan.doc	1 kb
3	File_example_AVI_480_750kB.avi	725 kb
4	Hasil_Video_LSB.avi	342.191.kb

4 KESIMPULAN

Berdasarkan pemaparan diatas dapat disimpulkan, yaitu algoritma kriptografi *AES* dapat digabungkan dengan metode steganografi yaitu *LSB* untuk menyandikan data file teks dan juga penyembunyian data *file* teks dalam media vidio yang dimplementasikan ke dalam *matlab.*

Dalam implementasi metode *AES* di *matlab* penyandian hanya dilakukan kepada data teksnya saja. Sedangkan penyisipan data *file* teks yang telah dienkripsi ke dalam vidio *.AVI* tidak dilakukan proses kompresi vidio sehingga ukuran awal vidio jauh lebih kecil dibandingan vidio yang sudah disisipkan data teks enkripsi. Hasil penelitian diharapkan dapat dikembangkan kembali agar *size* vidio *output* hasil penyisipan data enkripsi tidak terlalu besar.

Referensi

Abdullah, Dedy, and Doni Saputro. 2016. "Implementasi Algoritma Blowfish Dan Metode Least Significant Bit Insertion Pada Video Mp4." *Jurnal Pseudocode* 3(2): 137–45.

Amin, M. Miftakul. 2015. "Image Steganography Dengan Metode Least Significant Bit (Lsb)." *CSRID (Computer Science Research and Its Development Journal)* 6(1): 53.

Dista Amalia Arifah. 2011. "KASUS CYBERCRIME DI INDONESIA Indonesia's Cybercrime Case." *Jurnal Bisnis dan Ekonomi (JBE)* 18(2): 185–95.

Pabokory, Fresly Nandar, Indah Fitri Astuti, and Awang Harsa Kridalaksana. 2016. "Implementasi Kriptografi Pengamanan Data Pada Pesan Teks, Isi File Dokumen, Dan File Dokumen Menggunakan Algoritma Advanced Encryption Standard." *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer* 10(1): 20.

Prameshwari, Asri, and Nyoman Putra Sastra. 2018. "Implementasi Algoritma Advanced Encryption Standard (AES) 128 Untuk Enkripsi Dan Dekripsi File Dokumen." *Eksplora Informatika* 8(1): 52.

Rekamasanti, Farisah Qisthina, Ir Bambang Hidayat, and I Nyoman Apraz Ramatryana. 2015. "Implementasi Dan Analisis Video Steganografi Dengan Format Video Avi Berbasis Lsb (Least Significant Bit) Dan Ssb-4 (System of Steganography Using Bit 4) Implementation and Analysis of Steganography Avi Video Based on Lsb (Least Significant Bit) And." 2(2): 3129–36.