A Method Paper: Implementasi Kriptografi Untuk *File* Dengan Algoritma Aes (*Advanced Encryption Standard*) Rijndael Dan Algoritma Steganografi Lsb Pada Aplikasi Berbasis *Web*

Rd. Ridwan Permana Ruslan[1*], Setyawan Widyarto[1,2]

[1]Universitas Budli Luhur,
Jakarta, Indonesia
rdridwanpermana@gmail.com
[2]Centre for Gradutae Studies, Universiti Selangor

ABSTRAK

Isu keselamatan dan kerahsiaan data adalah sangat penting dalam sesebuah organisasi serta individu. Terutamanya jika data berada dalam rangkaian komputer yang disambungkan ke rangkaian awam seperti Internet. Sudah tentu, data yang sangat penting boleh dilihat atau dirampas oleh orang yang tidak diundang. Jika ini berlaku, data akan rosak atau bahkan hilang yang akan menyebabkan kerugian besar material. Aij (Advanced Enryption Standard) Rijndael adalah salah satu penyulitan yang sesuai untuk keselamatan data. Di samping itu, algoritma steganografi LSB adalah sangat sesuai untuk fail yang disimpan. Fungsi *hash* adalah algoritma yang menukarkan teks atau mesej ke dalam satu siri aksara rawak yang mempunyai beberapa aksara. Penyelidikan ini menerangkan pelaksanaan penyulitan fail AES-128bit dan algoritma kata laluan dan algoritma steganografi LSB pada media storan untuk penyimpanan fail dalam program berasaskan web menggunakan bahasa pengaturcaraan PHP. Pengujian keputusan aplikasi algoritma AES-128 dan LSB dilakukan dengan memasukkan teks dari fail yang disulitkan ke dalam fail asal dan membandingkan fail proses penyahsulitan dengan fail asal. Daripada keputusan ujian kesimpulan yang diperoleh adalah algoritma kriptografi AES-128 dan algoritma steganografi LSB berjaya dilaksanakan dalam program keselamatan data berasaskan web.

ABSTRACT

The issue of security and confidentiality of data is very important in an organization as well as individuals. Especially if the data is in a computer network that is connected / connected to a public network such as the internet. Of course, very important data can be seen or hijacked by uninvited people. Cause if this happens, our data will be damaged or even lost which will cause large material losses. Aij (Advanced Enryption Standard) Rijndael is one of the encryption that is suitable for data security. In addition, LSB steganography algorithm is a very good one for collected files. A hash function is an algorithm that converts text or messages into a series of random characters that have a number of characters. This research explains the implementation of AES-128bit file encryption and password algorithm and LSB steganography algorithm on storage media for file storage in web-based programs using the PHP programming language. Testing the results of the application of the AES-128 and LSB algorithm is done by inputting text from an encrypted file with the original file and comparing the decryption process file with the original file. From the test results obtained from the conclusions of the AES-128 cryptographic algorithm and LSB steganography algorithm successfully implemented in a web-based data security program.

Keywords: *Kriptografi, AES-128, Steganografi, LSB, PHP*

1. PENDAHULUAN

Keamanan suatu data merupakan faktor paling penting guna melindungi data tersebut. Untuk mengamankan data, digunakan berbagai macam cara dan teknik. Salah satunya adalah teknik kriptografi. Teknik kriptografi adalah sebuah teknik yang digunakan untuk melindungi data dimana teknik ini melakukan pengacakan terhadap isi data tersebut sehingga sulit untuk diartikan. Teknik ini sudah ada dan digunakan sejak jaman dahulu kala. Teknik ini biasanya digunakan ketika melakukan pengiriman pesan rahasia.

Teknik kriptografi menggunakan beberapa metode dalam penerapannya, mulai dari metode yang sederhana, hingga yang paling rumit dengan menggunakan rumus-rumus matematika. Dalam kriptografi, dikenal dua buah metode, yaitu: enkripsi dan dekripsi. Metode enkripsi digunakan untuk mengubah data asli atau disebut dengan *plaintext* menjadi sebuah data acak atau yang disebut dengan *cipher text*. Satu lagi ialah metode dekripsi. Metode dekripsi ialah sebuah metode yang digunakan untuk mengubah sebuah *ciphertext* menjadi *plaintext*. Metode ini merupakan kebalikan dari metode enkripsi.

2. LANDASAN TEORI

2.1 Kriptografi

Kriptografi berasal dari bahasa Yunani, yaitu: *cryptós* yang artinya "*secret*" (rahasia) dan *gráphein* yang artinya "*writting*" (tulisan). Jadi, kriptografi berarti "*secret writting*" (tulisan rahasia). Selain itu ada juga definisi lainyang dikemukakan oleh beberapa pakar kriptografi. Menurut pakar pertama, kriptografi adalah ilmu penulisan rahasia dengan tujuan menyembunyikan makna pesan (*cryptography is the science of secret writing with the goal of hiding the meaning of a message*) [4]. Menurut pakar kedua, kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentifikasi [3]. Ada juga definisi lain menurut pakar kriptografi ketiga yang mengemukakan bahwa kriptografi adalah ilmu mengenai teknik enkripsi dimana data diacak menggunakan suatu kunci enkripsi menjadi sesuatu yang sulit dibaca oleh seseorang yang tidak memiliki kunci dekripsi [2].

2.2 Tujuan Kriptografi

Ada empat tujuan mendasar dari ilmu kriptografi yang juga merupakan aspek keamanan informasi [5], yaitu:

a. Kerahasiaan atau privasi, adalah layanan yang digunakan untuk menjaga isi informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka informasi yang telah diberikan sandi.
b. Integritas data, berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain: penyisipan, penghapusan, dan penyubsitusian data lain ke dalam data yang sebenarnya.
c. Autentifikasi, berhubungan dengan identifikasi atau pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri.
d. Non-repudiasi, usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman atau terciptanya suatu informasi oleh yang mengirimkan atau membuat.

2.3 *Advanced Encryption Standard* (*AES*)

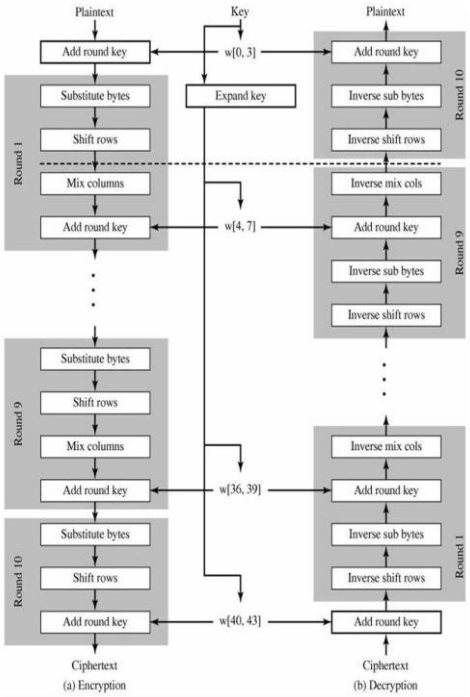
Advanced Encryption Standard (AES) merupakan algoritma *cryptographic* blok *chipertext* simetris yang paling banyak digunakan saat ini [4]. Algoritma AES mengunakan kunci kriptografi 128, 192, dan 256 *bits* untuk mengenkripsi dan mendekripsi data pada blok 128 *bits*. Algoritma kriptografi bernama AES Rijndael yang didesain oleh oleh Vincent Rijmen dan Joan Daemen asal Belgia keluar sebagai pemenang kontes algoritma kriptografi pengganti DES yang diadakan oleh NIST (*National Institutes of Standards and Technology*) milik pemerintah Amerika Serikat pada 26 November 2001. Algoritma Rijndael inilah yang kemudian dikenal dengan *Advanced Encryption Standard* (AES). Setelah mengalami beberapa proses standarisasi oleh NIST, Rijndael kemudian diadopsi menjadi *standard* algoritma kriptografi secara resmi pada 22 Mei 2002.

AES memiliki blok masukan dan keluaran serta kunci 128 bit. Untuk tingkat keamanan yang lebih tinggi, AES dapat menggunakan kunci 192 dan 256 bit. Setiap masukan 128 bit

plaintext dimasukkan ke dalam state yang berbentuk bujur sangkar berukuran 4×4 *byte*. State ini di-XOR dengan *key* dan selanjutnya diolah 10 kali dengan subtitusi-transformasi *linear-Addkey*. Dan diakhir, diperoleh *ciphertext*. Berikut ini adalah operasi Rijndael (AES) yang menggunakan 128 bit kunci:

a. Ekspansi kunci utama (dari 128 bit menjadi 1408 bit).
b. Pencampuran *subkey*.
c. Ulang dari i=1 sampai i=10 Transformasi: *ByteSub* (subtitusi per *byte*) *ShiftRow* (Pergeseren *byte* perbaris) *MixColumn* (Operasi perkalian GF (2) per kolom).
d. Pencampuran *subkey* (dengan XOR).
e. Transformasi : *ByteSub* dan *ShiftRow*.
f. Pencampuran *subkey*

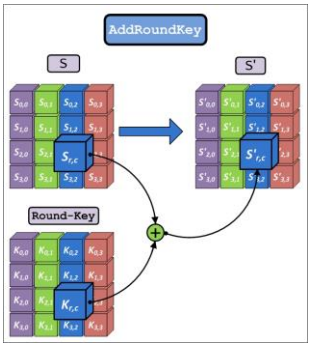
Secara umum metode yang digunakan dalam pemrosesan enkripsi dalam algoritma ini dapat dilihat melalui gambar 1.



Gambar 1: Diagram AES

a. Add Round Key

Add *Round Key* pada dasarnya adalah mengombinasikan *ciphertext* yang sudah ada dan *cipherkey* dengan hubungan XOR.



Gambar 2 : Add Round Key

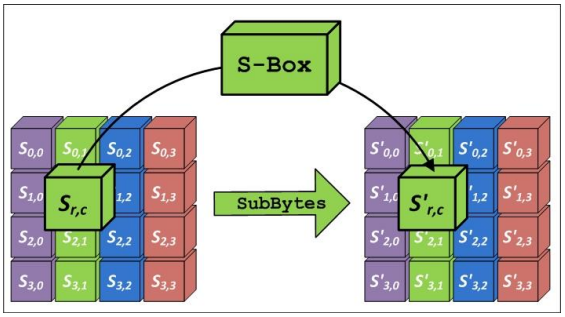
Pada gambar tersebut di sebelah kiri adalah chiperteks dan sebelah kanan adalah *round key*-nya. XOR dilakukan per kolom yaitu kolom-1 *ciphertext* di XOR dengan kolom-1 *round key* dan seterusnya.

b. Sub Bytes

Prinsip dari *Sub Bytes* adalah menukar isi matriks atau tabel yang ada dengan matriks atau tabel lain yang disebut dengan Rijndael S-*Box*. Di bawah ini adalah contoh *Sub Bytes* dan Rijndael S-*Box*.

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1x	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2x	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3x	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4x	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5x	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6x	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7x	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8x	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9x	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
ax	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
bx	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
cx	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
dx	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
ex	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
fx	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Gambar 3: Rijandel S-Box



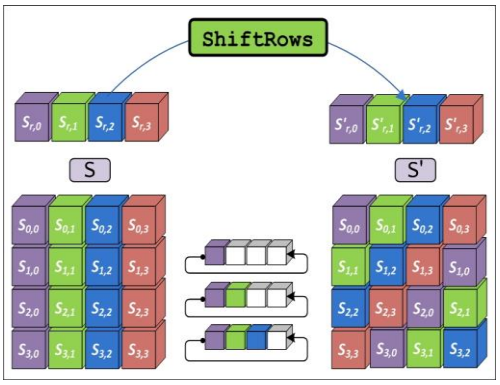
Gambar 4: Ilustrasi Sub Bytes

Gambar di atas adalah contoh dari Rijndael S-*Box* yang terdiri dari baris dan kolom. Seperti yang telah disebutkan

sebelumnya, tiap isi kotak dari blok chiper berisi informasi dalam bentuk heksadesimal yang terdiri dari dua digit, bisa angka-angka, angka-huruf, ataupun huruf-angka yang semuanya tercantum dalam Rijndael S-*Box*. Langkahnya adalah mengambil salah satu isi kotak matriks, mencocokkannya dengan digit kiri sebagai baris dan digit kanan sebagai kolom. Kemudian dengan mengetahui kolom dan baris, kita dapat mengambil sebuah isi tabel dari Rijndael S-*Box*. Langkah terakhir adalah mengubah keseluruhan blok chiper menjadi blok yang baru yang isinya adalah hasil penukaran semua isi blok dengan isi langkah yang disebutkan sebelumnya.

c. Shift Rows

Shift Rows, seperti namanya adalah sebuah proses yang melakukan *shift* atau pergeseran pada setiap elemen blok atau tabel yang dilakukan perbarisnya. Yaitu baris pertama tidak dilakukan pergeseran, baris kedua dilakukan pergeseran 1 *byte*, baris ketiga dilakukan pergeseran 2 *byte*, dan baris keempat dilakukan pergeseran 3 *byte*. Pergeseran tersebut terlihat dalam sebuah blok adalah sebuah pergeseran tiap elemen ke kiri tergantung berapa *byte* tergesernya, tiap pergeseran 1 *byte* berarti bergeser ke kiri sebanyak satu kali.



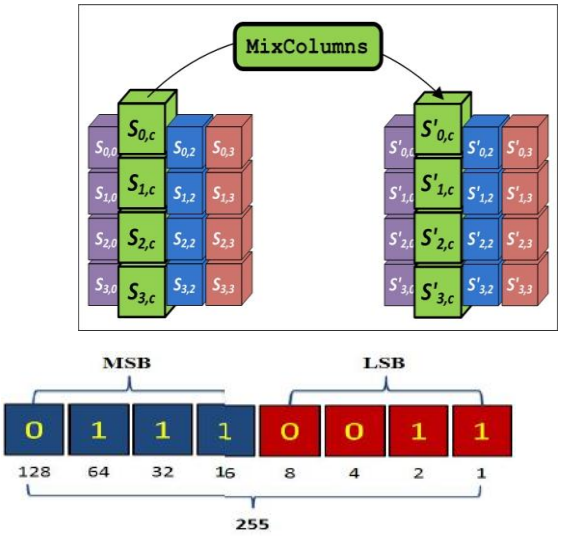
Gambar 5: Ilustrasi Dari Shift Rows

d. Mix Columns

Yang terjadi saat *Mix Column* adalah mengalikan tiap elemen dari blok chiper dengan matriks. Tabel sudah ditentukan dan siap pakai. Perkalian dilakukan seperti perkalian matriks biasa yaitu menggunakan *dot product*, lalu perkalian keduanya dimasukkan ke dalam sebuah blok chiper baru.

Tabel 1: Tabel Untuk Mix Columns

02	01	01	03
03	02	01	01
01	03	02	01
01	01	02	03



Gambar 6: Ilustrasi Dari Mix Columns

2.4 *Steganografi*

Sistem Steganografi akan menyembunyikan sejumlah informasi dalam suatu berkas dan akan mengembalikan informasi tersebut kepada pengguna yang berhak. Terdapat dua langkah dalam sistem Steganografi yaitu proses penyembunyian dan recovery data dari berkas penampung. Penyembunyian data dilakukan dengan mengganti bit-bit data di dalam segmen citra dengan bit-bit data rahasia. Metode yang paling sederhana adalah metode modifikasi LSB (*Least Significant Bit Modification*). Pada susunan bit di dalam sebuah byte (1 byte = 8 bit), ada bit yang paling berarti (*most significant bit* atau MSB) dan bit yang paling kurang berarti (*least significant bit* atau LSB)

Bit yang cocok untuk diganti adalah bit LSB, sebab perubahan tersebut hanya mengubah nilai *byte* satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya. Misalkan *byte* tersebut menyatakan warna merah, maka perubahan satu bit LSB tidak mengubah warna merah tersebut secara berarti. Lagi pula, mata manusia tidak dapat membedakan perubahan yang kecil. Misalkan segmen data citra sebelum perubahan:

Segmen data citra setelah ‘0 1 1 1‘

```
0 0 1 1 0 0 1 0  1 0 1 0 0 0 1 1
1 1 1 0 0 0 1 1  0 1 1 0 1 1 1 1
```

disembunyikan:

Ukuran data yang akan disembunyikan bergantung pada ukuran citra penampung. Pada citra 24-bit yang berukuran 256 ´ 256 pixel terdapat 65536 pixel, setiap pixel berukuran 3 byte (komponen RGB), berarti seluruhnya ada 65536 ´ 3 = 196608 byte. Karena setiap byte hanya bisa menyembunyikan satu bit di LSBnya, maka ukuran data yang akan disembunyikan di dalam citra maksimum. 196608/8 = 24576 byte Ukuran data ini harus dikurangi dengan panjang nama berkas, karena penyembunyian data rahasia tidak hanya menyembunyikan isi data tersebut, tetapi juga nama berkasnya. Semakin besar data disembunyikan di dalam citra, semakin besar pula kemungkinan data tersebut rusak akibat manipulasi pada citra penampung.

2.5 PHP (PHP*: Hypertext Preprocessor*)

PHP adalah singkatan dari "PHP: *Hypertext Preprocessor*", yang merupakan sebuah bahasa *scripting* yang terpasang pada HTML. Sebagian besar sintaks mirip dengan bahasa C, Java, dan Perl yang ditambah beberapa fungsi PHP yang spesifik. Tujuan utama penggunaan bahasa ini adalah untuk memungkinkan perancang *web* menulis halaman *web* dinamis dengan cepat. Halaman *web* biasanya disusun dari kode-kode html yang disimpan dalam sebuah *file* berekstensi .html. *File* html ini dikirimkan oleh *server* atau *file* ke *browser*.

Ada 4 kategori kunci yang membuat pengguna memiliki alasan khusus untuk menggunakan PHP [1], yaitu:

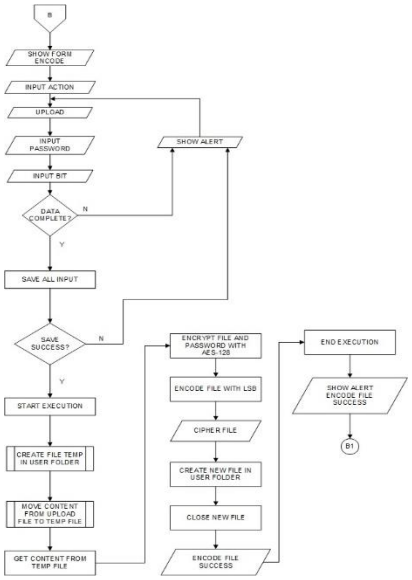
a. *Practically.*
b. *Power.*
c. *Possibility.*
d. *Price.*

3. RANCANGAN SISTEM DAN APLIKASI

Tahap-tahap yang terjadi dalam skema proses sistem aplikasi ini adalah sebagai berikut:

3.1 Skema Proses *Encode*

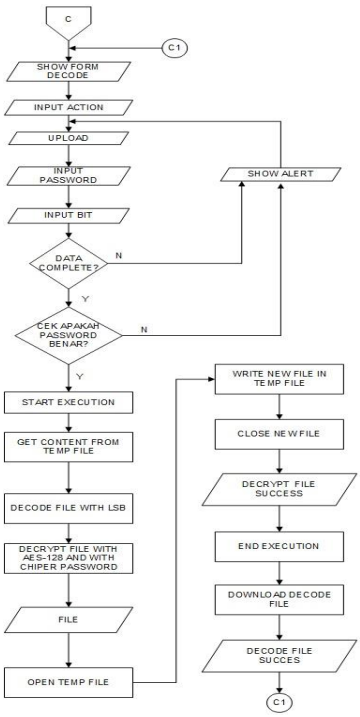
Encode merupakan proses pembuatan *cipher file* dengan teknik kriptografi. Proses ini memerlukan *input*-an berupa *file* serta panjang bit. Kunci rahasia (*password*) akan diberikan otomatis oleh sistem dengan nilai yang acak. *Output* yang dihasilkan pada proses ini adalah sebuah *cipher file* hasil proses *encode*.



Gambar 8: Skema Proses Encode

3.2 Skema Proses *Decode*

Decode merupakan proses pengembalian data dari *cipher file* menjadi data asli. Proses ini memerlukan *input*-an berupa *file*, *password*. *Output* yang dihasilkan pada proses ini adalah sebuah *plain file* atau data asli.



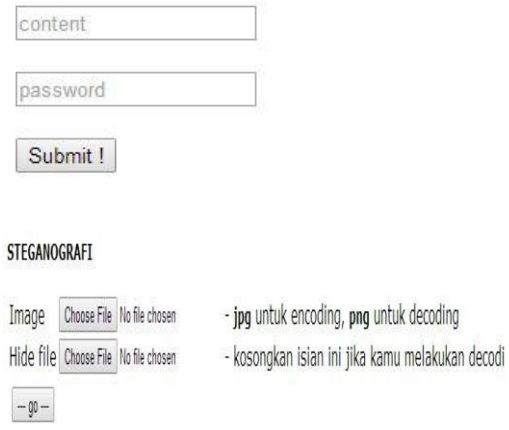
Gambar 9: Skema Proses Decode

4. HASIL DAN PEMBAHASAN

4.1 Tampilan Layar Program

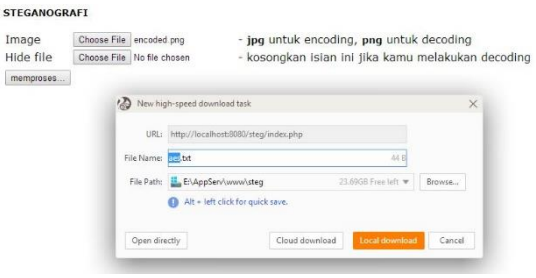
Tampilan layar program hasil *coding* yang telah disesuaikan dengan kebutuhan *user* yang telah ditentukan pada awal rancangan.

4.2 Tampilan Layar Menu *Encode File*



Gambar 13: Tampilan Layar Menu Encode File

4.3 Tampilan Layar Menu *Decode File*



Gambar 14: Tampilan Layar Menu Decode File

4.4 Penggunaan Program

a. Pengujian Proses *Encode*

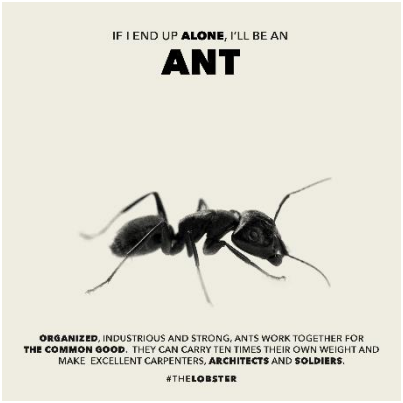
File image

File hide

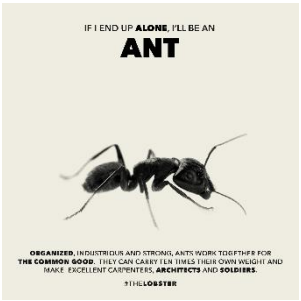
File hasil

Gambar 15: Hasil *Encode*

b. Pengujian Proses *Decode*



File hide



File image

Encrypred: 4IPS5U86esNHX+EtqkNOGlqoj6xp2t6BJDZMV7CLLMw=
go to stegeno

File hide

Decrypred: ridwan

Gambar 16: Hasil *Decode*

5 KESIMPULAN

5.2 Kesimpulan

Setelah melalui tahap proses pembuatan dan pengujian program Tugas ini, dapat disimpulkan beberapa hal, yaitu:

a. Program yang dikembangkan dapat digunakan dengan baik untuk proses *encode* dan *decode* berbagai macam jenis *file* (*document, image)*.
b. Untuk mengetahui isi dari *file* hasil *encode*, harus melakukan proses *decode* dan mengetahui *password*.
c. Proses *decode* tetap bisa dijalankan walaupun tidak mengetahui *password* pada proses *encode file*, namun *file* hasil *decode* masih tetap tidak bisa dibaca karena masih berupa data acak.
d. Dengan adanya program ini, maka selain penyimpanan *file* yang terjamin, isi dari *file* juga bisa menyembunyikan Ike

5.3 Saran

Beberapa saran yang penulis dapat berikan untuk pengembangan lebih lanjut antara lain:

a. Untuk *file* yang berukuran relatif besar sebelum proses enkripsi, akan lebih baik apabila dikompresi terlebih dulu, hal ini berguna untuk mempercepat proses *encode* dan *decode*.
b. Untuk lebih mempermudah penggunaan program, sebaiknya ada penambahan modul yang dapat memberikan fasilitas *multiple file upload* pada *form encode* dan *decode* agar pada saat satu kali proses *encode* ataupun *decode* bisa memproses lebih dari satu *file*.
c. Jika memberikan fasilitas *multiple file upload* pada *form encode* dan *decode* sebaiknya diperhatikan maksimal *file* yang boleh di-*upload* mengingat waktu ekseskusi *file* yang akan bertambah lama karena banyaknya *file* yang di-*upload*-kan.

DAFTAR PUSTAKA

[1] Gilmore, W. Jason 2010, *Beginning PHP and MySQL From Novice to Professional*, United States of America, Springer.
[2] Kromodimoeljo, Sentot 2010, *Teori & Aplikasi Kriptografi*, Indonesia, SPK IT Consulting.
[3] Munir, Rinaldi 2006, *Kriptografi*, Bandung, Informatika.
[4] Paar, Christof, & Jan Pelzl 2010, *Understanding Cryptography*, New York, Springer.
[5] Paul, Gautham, & Subhamoy Maitra 2012, *Steganography It's Variants*, United