APLIKASI CREDENTIAL LOGIN DENGAN METODE STEGANOGRAFI "LSB" (LEAST SIGNIFICANT BIT) DAN ALGORITMA KRIPTOGRAFI "VIGENERE"

Novia Busiarli[1)], Yuli Kurniawati[2)], Akrilvalerat Deainert Wierfi[3)]

[1), 2),3)] Teknik Informatika STMIK AMIKOM Yogyakarta
Jl Ring road Utara, Condongcatur, Sleman, Yogyakarta 55281
Email : liviaqueen11@gmail.com[1)], lilikurniiaa@yahoo.com[2)], akrilvha8@gmail.com [3)]

Abstrak

Pentingnya sebuah aplikasi untuk mengidentifikasi akses pengguna adalah suatu upaya mencegah penyusup untuk melakukan penyerangan terhadap sistem aplikasi yang digunakan. Sebuah sistem aplikasi dikatakan layak pakai jika sudah mengandung aspek-aspek kemananan komunikasi komputer. Credential login merupakan salah satu cara mengidentifikasi pengguna dalam penggunaan sistem, banyak metode yang dapat di lakukan dalam hal pemfilteran terkait Credential login ini, diantaranya identifikasi dengan penggunaan sertifikat login, otentikasi menggunakan token-token dsb.

Dalam hal ini, penulis mencoba untuk membuat aplikasi Credential login sederhana dengan metode super enkripsi yaitu penggunaan Steganografi (LSB) dan Kriptografi Vigenere. Saat User mengentrikan Password (plaintext) maka sistem dengan otomatis mengenkripsi plaintext tersebut menjadi sebuah chipertext yang nantinya akan disisipkan pada sebuah gambar yang kita sebut Steganografi (image). Jika suatu saat User lupa dengan Password maka sistem akan mendeksripsikan image tsb dengan kunci yang telah ditentukan dengan metoda Kriptografi Vigenere.

Dari penelitian yang dilakukan, penulis dapat menarik kesimpulan bahwa, penggunaan Steganografi LSB tidak mempengaruhi kualitas sebuah gambar dan dapat menampung pesan yang tersimpan di dalamnya. Hanya saja terjadi perubahan size gambar / image saat enkripsi dari ukuran gambar sebelumnya. Pada penelitian ini juga memiliki kelemahan dalam penggunaan Steganografi sebagai Credential login dibandingkan dengan penggunaan sertifkat atau pun token – token.

Kata kunci: *Credential, Steganografi LSB, Kriptografi Vigenere, Password, User.*

1. Pendahuluan

Menurut penelitian yang dilakuan oleh Basuki Rahmat NF dalam jurnalnya yang berjudul " Steganografi Menggunakan Metode Least Significant Bit Dengan Kombinasi Algoritma Kriptografi Vigenere dan RC4" menyimpulkan bahwa Algoritma kriptografi klasik dan modern, yaitu Vigenère dan RC4 dapat dikombinasi-kan dalam sebuah sistem untuk memberikan dua lapis proteksi dalam menyembunyi-kan pesan rahasia. Teknik steganografi dengan metode Least Significant Bit (LSB) untuk menyembunyi-kan pesan teks pada gambar bitmap dapat diaplikasikan menggunakan Delphi 2010. Kriptografi dan steganografi dapat diintegrasikan menjadi satu dalam sebuah sistemaplikasi. Pesan teks terlindungi dengan algoritma kriptografi dan tersembunyi dalam sebuah gambar[1].

Menurut penelitian yang dilakuan oleh Krisnawati yang berjudul "Metode least significant bit (lsb) dan end of file (eof) Untuk Menyisipkan Teks kedalam Citra Grayscale" menyimpulkan bahwa Penyisipan pesan/data ke dalam citra dapat dilakukan dengan metode LSB dan EOF. Metode LSB akan mengganti bit terakhir kode biner masing-masing piksel. Kelebihan dari metode ini adalah ukuran citra tidak berubah/tetap, sehingga tidak mengakibatkan kecurigaan akan adanya pesan rahasia dalam citra. Kekurangan metode ini adalah jumlah karakter pesan yang disisipkan terbatas, sehingga besarnya citra harus menyesuaikan besarnya pesan yang dikirim. Metode EOF akan meletakkan pesan di akhir citra sehingga ukuran file akan bertambah besar, oleh karena itu pesan teks yang disisipkan tidak terbatas jumlahnya[2].

Teknologi informasi yang berkembang pesat membantu pekerjaan manusia menjadi lebih mudah. Diharapkan dengan kemudahan yang diberikan selaras dengan keamanan yang di inginkan. Sebuah sistem dikatakan aman jika sudah memenuhi aspek-aspek keamanan dalam komunikasi komputer. Salah satu keamanan tersebut dapat dilihat pada *authorithy User*. Ini dapat dilihat pada management hak *access User*, sebuah sistem terdapat fasilitas untuk mengidentifikasi hak *access* pengguna dalam sebuah *credential login*. Banyak metode yang dipakai untuk mengimplementasikan *credential login* ini yang terintegrasi dengan sistem lainnya seperti keterkaitan antara *authority access* dengan konfirmasi menggunakan *email*.

Dalam penelitian ini, penulis mencoba membuat sebuah inovasi dengan pemanfaatan *image* / gambar sebagai otentikasi dari *authority access user* tersebut. Dalam hal ini penulis membahas bagaimana penyandian *password* (*plaintext*) menggunakan metode *Steganografi* dan *Kriptografi Vigenere*. Tujuannya agar sebuah aplikasi dapat diakses oleh *User* yang tepat, artinya hanya *User*

yang terregistrasi dan dapat melakukan *login* dengan pemilihan *password* dan *image* yang terecord di sistem. Karena sistem akan melakukan pencocokan *image* dan *password* saat melakukan *login* dengan *image* dan *password* yang terrecord di *Database*.

Kriptografi berasal dari bahasa yunani, *kripto* berarti *secret* (rahasia) dan *grafi* berarti *writing* (tulisan). Menurut terminologynya *Kriptografi* adalah ilmu dan seni untuk menjaga keamanan pesan ketika dikirim dari suatu tempat ke tempat lannya[3]. *Kriptografi* disebut juga seni dan ilmu untuk menjaga keamanan sebuah pesan[4] dengan cara melakukan enkripsi dan dekripsi. Enkripsi merupakan teknik untuk membuat pesan menjadi tidak dapat dibaca (*chipertext*) sedangkan dekripsi merupakan kebalikan dari enkripsi yaitu teknik merubah *chipertext* mejadi *plaintext* dengan menggunakan kunci[4].

Terjadinya pertukaran informasi akan mengakibatkan terjadinya ancaman keamanan / pencurian informasi oleh pihak-pihak yang tidak bertanggung jawab, berikut ancaman keamanan yang terjadi dalam perukaran informasi [4]:

a. *Interruption*
   Merupakan ancaman terhadap availability, informasi, data yang ada dalam *system computer* dirusak, dihapus, sehingga tidak ada data lagi jika suatu saat dibutuhkan [4].
b. *Interception*
   Merupakan ancaman terhadap kerahasiaan informasi yang ada disadap atau orang yang tidak berhak mendapat akses ke komputer dimana informasi tersebut disimpan [4].
c. *Modification*
   Merupakan ancaman terhadap integritas orang yang tidak berhak berhasil menyadap lalu lintas informasi yang sedang dikirim dan dirubah sesuai dengan keinginan orang tersebut [4].
d. *Fabrication*
   Merupakan ancaman terhadap integritas orang yang tidak berhak meniru (memalsukan) suatu informasi yang ada sehingga orang yang menerima informasi tersebut menyangka informasi tersebut berasal dari orang yang dikehendaki oleh sipenerima informasi tersebut [4].

Agar aman dalam berkomunikasi lewat jaringan internet maka diharapkan adanya aspek-aspek keamanan aga terhindar dari berbagai ancaman tersebut, adapun aspek-aspek tersebut adalah:

a. *Authentication*
   Layanan keamanan jaringan yang memberikan kepastian terhadap identitas sebuah entitas yang terlibat dalam komunikasi data [3].
b. *Integrity*
   Layanan keamanan jaringan yang memastikan bahwa data yang diterima oleh penerima adalah benar-benar sama dengan data yang dikirim oleh pengirim [3].
c. *Nonrepudation*
   Layanan keamanan jarinagn yang menghindari penolakan atas penerimaan / pengiriman data yang telah terkirim [3].
d. *Authority*
   Informasi yang berada pada system jaringan tidak dapat dimodifikasi oleh pihak yang tidak berhak atas akses tersebut [5].
e. *Confidentiality*
   Layanan keamanan jaringan yang memproteksi data tertransmisi terhadap pengungkapan oleh pihak yang tidak berwenang [3].
f. *Privacy*
   Lebih kearah data-data yang sifatnya *privat* (pribadi) [5].
g. *Availability*
   Layanan sistem yang membuat sumber daya sistem tetap dapat diakses dan digunakan ketika ada permintaan dari pihak yang berwenang [3].
h. *Access* Control
   Layanan keamanan jaringan yang menghalangi penggunaan tidak terotorisasi terhadap sumber daya [3].

Kriptografi Vigenere merupakan *Kriptografi* subsitusi *poli-alfabetik* klasik yang terkenal [4]. *Kriptografi Vigenere* merupakan system sandi *poli-alfabetik* yang mengenkripsi sekaligus sebuah teks yang terdiri dari beberapa huruf dan menggunakan subsitusi dengan fungsi *shift* [3].

Contoh:

P_1	B	U	L	A	N	P	U	R	N	A	M	A
K	7	10	21	7	10	21	7	10	21	7	10	21

C[0] =1 + 7 mod 26= 8 C[1] = 20 + 10 mod 26 = 4
C[2] =11 + 21 mod 26= 6 C[3] = 0 + 7 mod 26 = 7
C[4] =13 + 10 mod 26= 23 C[5] =15+21 mod 2 =10
C[6] =20 + 7 mod 26= 1 C[7] = 17 + 10 mod 26 = 1
C[8] =13 + 21 mod 26= 8 C[9] = 0 + 7 mod 26 = 7
C[10]=12 + 10 mod 26= 22 C[11] =0 + 21 mod 26 = 21

Sehingga *chipertext*: “IEGHXKBBIHWV”

Teknik subsitusi *Vigenere* bisa dilakukan dengan 2 cara [5]:

a. 1. Angka
   Teknik subsitusi *Vigenere* dengan menggunakan angka dengan menukarkan huruf dengan angka. Hal ini hampir sama dengan *shift cipher.*

Tabel 1.*Tabel Subsitusi angka Vigenere*

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

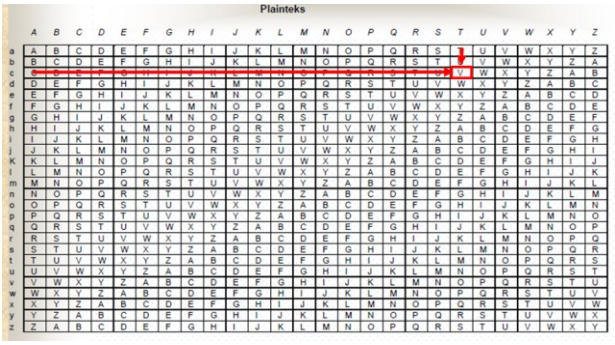
Contoh:

T	H	I	S	C	R	Y	P	T	O	S	Y	S
19	7	8	18	2	17	24	15	19	14	18	24	18
2	8	15	7	4	17	2	8	15	7	4	17	2
21	15	23	25	6	8	0	23	8	21	22	15	20

T	E	M	I	S	N	O	T	S	E	C	U	R	E
19	4	12	8	18	13	14	19	18	4	2	20	17	4
8	15	7	4	17	2	8	15	7	4	17	2	8	15
1	19	19	12	9	15	22	8	25	8	19	22	25	19

Plaintext : this cryptosystem is not secure
Kunci : chiper
Chipertext : vpxzgiaxivwpubttmjpwizitwzt

a. 2. Huruf
Teknik subsitusi *Vigenere* dengan menggunakan huruf, bisa digunakan gambar 1 dibawah ini:



Gambar 1. *Table subsitusi huruf Vigenere*

Plaintext : this cryptosystem is not secure
Kunci : chiper
Chipertext : vpxzgiaxivwpubttmjpwizitwzt

Steganografi adalah ilmu menyembunyikan teks pada media lain yang telah ada sedemikian sehinggan teks yang tersembunyi menyatu dengan media itu [3]. Media tempat penyimpanan pesan tersembunyi dapat berupa media teks, gambar, audio, video. *Steganografi* yang kuat memiliki sifat media yang telah tertanam teks tersembunyi sulit dibedakan dengan media asli namun teks tersembunyi tetap data diekstraksi.

Contoh sederhana teknik *Steganografi* pada media gambar misalnya dengan mengubah nilai LSB (*least significant bit*). Sebagai contoh huruf A akan disisipkan ke dalam citra berukuran 3 x 3 *pixel*, misalkan data *pixel*-nya adalah sebagai berikut:

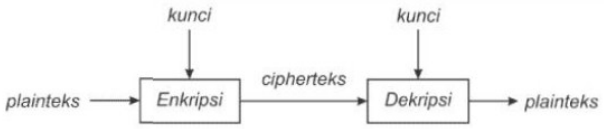
(01100101 10101100 10011010)
(01000101 11001010 11001101)
(11110000 01110101 10100011)

Sedangkan representasi *biner* huruf A adalah 01000001. Dengan menyisipkan bit-bit huruf A tersebut pada data *pixel* diatas, maka setiap bit dari huruf A tersebut akan menggantikan posisi *LSB* dari data *pixel* citra menjadi:

(01100100 10101101 10011010)
(01000100 11001010 11001100)
(11110000 01110100 10100011)

2. Pembahasan

Seperti yang kita ketahui, kriptografi memiliki dua konsep utama, yaitu enkripsi (*encryption*) dan dekripsi (*decryption*). Enkripsi adalah proses penyandian *plainteks* menjadi *cipherteks*, sedangkan dekripsi adalah proses mengembalikan *cipherteks* menjadi *plainteks* semula. Enkripsi dan dekripsi membutuhkan kunci sebagai parameter yang digunakan untuk transformasi. Gambar 2 memperlihatkan skema enkripsi dan dekripsi dengan menggunakan kunci.



Gambar 2.*Diagram flow proses enkripsi & dekripsi*

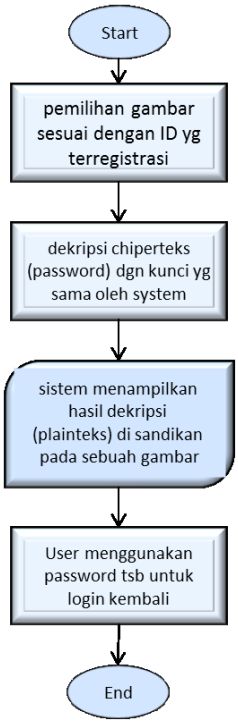
Didalam melakukan penelitian, penulis terlebih dahulu membuat konsep untuk *enkripsi* menggunakan *Kriptografi Vigenere* dan *Steganografi*. Penulis merancang beberapa *form* (*User Interface*) sebagai media *input* / *outputnya*. Sedangkan untuk algoritma proses *enkripsi* dan *steganografi* tersebut akan dijelaskan seperti gambar 3 Bagan / alur kerja berikut:

Gambar 3.*Diagram flow proses enkripsi*

Gambar diatas menerangkan proses *enkripsi* sebuah *plaintext* (*Password*). Ketika *User* mengentrikan sebuah *plaintext* pada kolom *Password* maka sistem akan mengesnkripsikan *plaintext* tersebut menjadi sebuah *chipertext* dengan sebuah kunci. Dimana kunci masing-masing id berbeda-beda artinya setiap ID *User* yag teregistrasi meregistrasikan kunci untuk enkripsi *Password* dan dekripsi *Passwordnya*.

Hasil enkripsi (*chipertext*) disisipkan pada sebuah gambar yang dipilih oleh *User* dan *User* akan diminta memilih gambar untuk penyisipan *chiperteks* tadi. Kemudian gambar yang sudah terenkripsi disimpan dengan ID *User* yang sama. Akan ada 2 buah gambar yang tersimpan dengan bentuk yang serupa tetapi memiliki perbedaan ukuran dan nama file gambar untuk membedakan mana gambar orisinil dengan gambar yang sudah terenkripsi.

Sedangkan untuk proses dekripsi akan dijelaskan seperti pada gambar 4 bagan / alur dekripsi berikut ini:

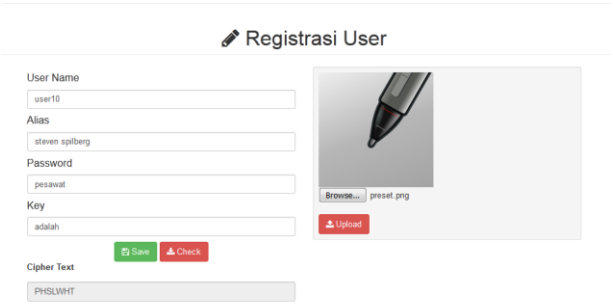


Gambar 4.*Diagram flow proses dekripsi*

Gambar diatas menjelaskan proses dekripsi dari *cipherteks* yang tersisipkan pada gambar. Untuk dekripsi *plaintext* ini, *User* akan memilih gambar yang terenkripsi sebelumnya dengan ID masing-masing user. Kemudian *User* mengentrikan kunci yang sama saat me-enkripsikan *plaintext* sebelumnya. Sehingga sistem memproses dan menampilkan kembali *Password* tersebut dan *User* dapat menggunakan *Password* itu kembali saat melakukan *login*.

2. 1 Enkripsi Kriptografi Vigenere

Penulis merancang sebuah *form* untuk melakukan registrasi seperti gambar 5 berikut:



Gambar 5. *User Interface Form Registrasi User*

Gambar diatas merupakan rancangan sistem dimana *End User* diminta untuk mengisikan *User Name* (kombinasi huruf dan angka). *User Name* ini sebagai identitas yang digunakan saat proses login kemudian Alias yang menjadi identitas umum *User* agar mudah diingat. Selanjutnya *End User* diminta untuk mengentrikan *Password* dan kunci kemudian sistem akan merecord data dalam satu ID.

Setelah itu *End User* akan diminta bantuan untuk mengklik *button check* untuk menampilkan *chipertext* hasil dari enkripsi. Berikut penjelasan enkripsi *plaintext*

Plaintext : PESAWAT

Kunci : ADALAH

dengan menggunakan table subsitusi angka

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Maka enkripsi untuk *plaintext* “PESAWAT” dengan kunci “ADALAH”adalah:

	P	E	S	A	W	A	T
Pi	15	4	18	0	22	0	19
	A	D	A	L	A	H	A
K	0	3	0	11	0	7	0
C_i	15	7	18	11	22	7	19
	P	**H**	**S**	**L**	**W**	**H**	**A**

Dengan operasi *shift* didapat seperti berikut:

C[0] = 15 + 0 mod 26 = 15 → **P**
C[1] = 4 + 3 mod 26 = 7 → **H**
C[2] = 18 + 0 mod 26 = 18 → **S**
C[3] = 0 + 11 mod 26 = 11 → **L**
C[4] = 22 + 0 mod 26 = 22 → **W**
C[5] = 0 + 7 mod 26 = 7 → **H**
C[6] = 19 + 0 mod 26 = 19 → **A**

2. 2 Enkripsi Steganografi LSB

Hasil *chipertext* ini akan disisipkan kedalam sebuah gambar. Disini kita menggunakan gambar dengan ukuran 200 x 200 *pixel*. Masing-masing *pixel* akan diuraikan berdasarkan bineri *grey level* (*RGB*). Dari ukuran gambar didapat matrix dari masing-masing *grey level* seperti berikut:

permisalan, sampling 1 x 8 pixel dimana matriks RGB dari 1 x 8 pixel tsb adalah:

[239 239 239 239 239 239 239 238]
RGB[0][0] = [239 239 239]
avg[0][0] = (R+G+B)/3
= (239+239+239)/3 = 239

biner [0][0] = 11101111

RGB[0][1] = [239 239 239]
avg[0][1] = (R+G+B)/3
= (239+239+239)/3 = 239
biner [0][1] = 11101111

RGB[0][2] = [239 239 239]
avg[0][2] = (R+G+B)/3
= (239+239+239)/3 = 239
biner [0][2] = 11101111

RGB[0][3] = [239 239 239]
avg[0][3] = (R+G+B)/3
= (239+239+239)/3 = 239
biner [0][3] = 11101111

RGB[0][4] = [239 239 239]
avg[0][4] = (R+G+B)/3
= (239+239+239)/3 = 239
biner [0][4] = 11101111

RGB[0][5] = [239 239 239]
avg[0][5] = (R+G+B)/3
= (239+239+239)/3 = 239
biner [0][5] = 11101111

RGB[0][6] = [239 239 239]
avg[0][6] = (R+G+B)/3
= (239+239+239)/3 = 239
biner [0][6] = 11101111

RGB[0][7] = [238 238 238]
avg[0][7] = (R+G+B)/3
= (238+238+238)/3 = 238
biner [0][7] = 11101110

Sebelumnya kita sudah mendapatkan *chipertext* dari hasil enkripsi. *Chipertext* tersebut akan kita uraikan juga binery nya sehingga didapat lah seperti berikut ini:

01010000 01001000 01010011 01001100 01010111 01001000 01010100, dimana P = **01010000**

Setiap binery dari chipertext ini akan disisipkan pada binery masing-masing pixel gambar sehingga terjadi perubahan binery gambar seperti berikut:

P = 1110111**0** 1110111**1** 1110111**0** 1110111**1** 1110111**0** 1110111**0** 1110111**0** 1110111**0**

Dan begitu untuk *pixel-pixel* dan *chipertext* seterusnya.

2. 3 Dekripsi Kriptografi Vigenere

Jika *User* lupa akan *Password*, sistem akan membantu *User* untuk mendapatkan *Password* nya kembali dengan cara mendekripsikan gambar yang sudah terenkripsi sebelumnya. Proses dekripsi dilakukan dengan menggunakan kunci yang sama seperti penjelasan berikut:

Chipertext :

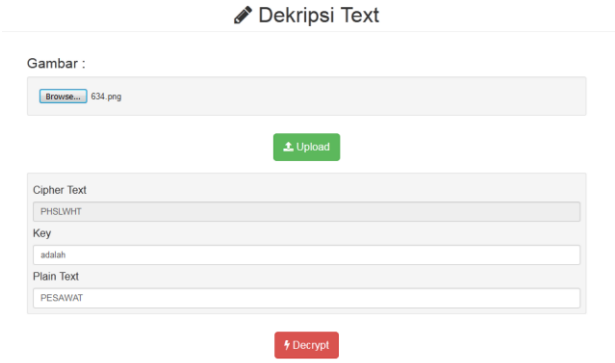
P H S L W H A

Kunci : ADALAH

Plaintext :

C[0] = 15 - 0 mod 26 = 15 → **P**
C[1] = 7 - 3 mod 26 = 4 → **E**
C[2] = 18 - 0 mod 26 = 18 → **S**
C[3] = 11 - 11 mod 26 = 0 → **A**
C[4] = 22 - 0 mod 26 = 22 → **W**
C[5] = 7 - 7 mod 26 = 0 → **A**
C[6] = 19 - 0 mod 26 = 19 → **T**

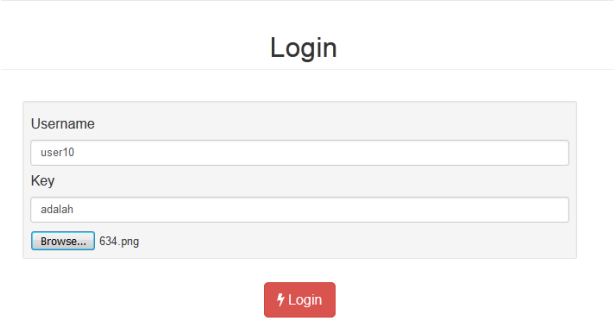
Sehingga terlihat *plaintext* pada *user interface* dengan perhitungan manual seperti gambar 6 berikut ini:



Gambar 6. *User Interface Form Dekripsi Password*

Gambar diatas menerangkan bahwa *End User* memilih gambar sesuai dengan ID-nya yang terregigstrasi kemudian upload gambar tersebut. Ketika itu juga secara otomatis sistem menampilkan *chipertext* yang terenkripsi sebelumnya.

Kemudian *End User* mengentrikan kunci yang sama saat pertama kali mengenkripsikan *plaintext* sehingga sistem memproses dan menampilkan kembali *plaintext* atau *Password* yang diinginkan *User*. Kemudian *User* diminta lagi untuk *login* seperti gambar 7 berikut ini:

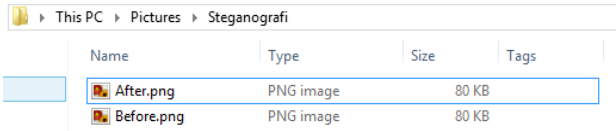


Gambar 7. *User Interface Form Login*

Gambar diatas menerangkan perbedaan *form login* konvesional ke inovasi dari *form login* yang penulis rancang. Disini penulis menggunakan *Username* dan *key* serta *image* yang sudah terregistrasi. *Field-field* yang tertera di *form login* ini akan dicocokan dengan data yang sudah ada di *Database*. *Login* akan berhasil dilakukan jika data yang dentrikan *User* sama dengan data yang sudah ada di *Database*.

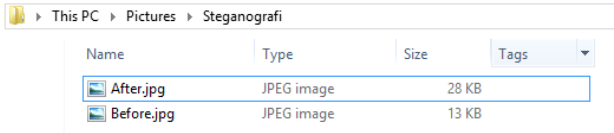
2. 4 Hasil *Steganografi* pada gambar

Setelah dilakukan enkripsi pada gambar dengan format *PNG* dan *JPG*, dimana enkripsi gambar dengan format *PNG* tidak terdapat perbedaan ukuran *file* yang *significant* seperti pada gambar berikut:



Gambar 8. *Hasil Perbandingan Ukuran file gambar dengan format PNG*

sedangkan enkripsi gambar dengan format JPG terdapat perbedaan ukuran gambar hampir 2 X ukuran gambar semula seperti pada gambar berikut ini:



Gambar 9. *Hasil Perbandingan Ukuran file gambar dengan format JPG*

3. Kesimpulan

Dari penelitian yang penulis lakukan di dapat kesimpulan:

1. Penggunaan *Steganografi* (*image*) menggunakan metode *Least Significat Bit* (LSB) tidak mempengaruhi kualitas *image*. Hanya saja terdapat perubahan ukuran *file* yang tidak terlalu significant untuk format file PNG / BMP. Pengecualian format file JPG akan terjadi perubahan size gambar hingga mencapai 2 x dari ukuran sebelumnya.
2. *User* akan dengan mudah mendapatkan *Passwordnya* kembali setelah melakukan proses dekripsi dari gambar yang terenkripsi sebelumnya. Karena disini tidak ada pemanfaatan *email* untuk mendapatkan *Password* kembali.
3. *Password* hanya akan didapat kembali jika *User* memilih gambar dan *key* nya dengan tepat dan benar. Ini merupakan salah satu trik mengelabui penyusup untuk tidak melakukan *hacking* terhadap sistem. Karena di batasi oleh *Username, key* dan *image* yang ada pada *form login*.

Daftar Pustaka

[1] Basuki Rahmat, N. F. (2010). *Steganografi Menggunakan Metode Least Significant Bit Dengan Kombinasi Algoritma Kriptografi Vigenère Dan Rc4*. Jurnal Dinamika Informatika, 1-17.

[2] Krisnawati. (2008). *Metode least significant bit (lsb) dan end of file (eof)* Untuk Menyisipkan Teks kedalam Citra Grayscale. Seminar Nasional Informatika 2008 (semnasIF 2008), 1-6

[3] Sadikin. Rifki, *Kriptografi Untuk Keamanan Jaringan*, Yogyakarta: Andi Offset Yogyakarta, 2012.

[4] Kurniawan. Yusuf, *Kriptografi Keamanan Internet dan Jaringan Komunikasi*, Bandung: Informatika, 2004.

[5] Ariyus. Dony, *Kriptografi Keamanan Data dan Komunikasi*, Yogyakarta: Graha Ilmu, 2006.

Penulis

Novia Busiarli, memperoleh gelar Ahli Madya Komputer (AMd), Jurusan Teknik Informatika Politeknik Negeri Padang, lulus tahun 2011. Saat ini sedang melanjutkan pendidikan ke jenjang Strata-1 di STMIK AMIKOM Yogyakarta.

Yuli Kurniawati, memperoleh gelar Ahli Madya Komputer (AMd), Jurusan Teknik Informatika Universitas Muhammadiyah Magelang , lulus tahun 2013. Saat ini sedang melanjutkan pendidikan ke jenjang Strata-1 di STMIK AMIKOM Yogyakarta.

Akrilvalerat Deainert Wierfi, memperoleh gelar Ahli Madya Komputer (AMd), Jurusan Teknik Informatika Politeknik Negeri Pontianak, lulus tahun 2015. Saat ini sedang melanjutkan pendidikan ke jenjang Strata-1 di STMIK AMIKOM Yogyakarta.