Implementasi Hidden Text dengan Algoritma Blowfish dan Kompresi Huffman dalam Security Dokumen

Siti Muryanah[1*], Diah Rahmawati[2]

[1,2] Program Studi Teknik Informatika, Fakultas Teknik, Universitas Islam Syekh Yusuf
Jl. Syekh Yusuf No. 10, Kota Tangerang, Banten, Indonesia
Email: *siti.muryanah@unis.ac.id

Abstrak. Penyimpanan data baik berupa dokumen atau lainnya tanpa adanya pengamanan apapun, dapat menimbulkan pencurian data oleh orang yang tidak bertanggung jawab.mKriptografi dan steganografi sebagai salah satu cara untuk menjaga keamanan informasi tersebut. Informasi yang kita miliki (*plaint text*) akan diubah menjadi suatu informasi acak atau sulit dimengerti (*chiper text*), biasa disebut dengan enkripsi. Metode enkripsi dalam kriptografi yang digunakan dalam penelitian ini adalah metode blowfish dimana panjang blok tetap yaitu 64 bit serta ukuran kunci antara 32 bit sampai 448 bit. Informasi acak hasil enkripsi akan disembunyikan atau disisipkan ke dalam citra digital (*encode*) sehingga tidak menimbulkan orang lain curiga bahwa dalam citra digital tersebut terdapat suatu informasi yang sedang disembunyikan. Algoritma steganografi dalam penelitian ini adalah *least significant bit (LSB)*, penyisipan pada bit paling kanan atau bit rendah sehingga citra digital yang telah disisipi suatu informasi rahasia (*stego image*), secara kasat mata manusia tidak ada perbedaannya baik sebelum atau sesudah disisipi informasi rahasia. Hasil penelitian ini, berhasil mengamankan dokumen dengan mengunakan kriptografi, steganografi dan kompresi Huffman dengan format pdf, xlsx, docx, pptx dan txt. Ukuran dokumen hasil proses enkripsi menjadi lebih kecil dengan proses kompresi Huffman, dengan nilai rata-rata 57% dari dokumen aslinya. Pengujian mengunakan 3 file dan 3 citra yang berbeda dalam masing-masing format file yaitu pdf, xlsx, docx, pptx dan txt. Pengujian terhadap stego image menghasilkan kualitas tinggi dengan nilai MSE 0,092 dan PSNR 57,12 dB.

Kata kunci: *blowfish; enkripsi; kriptografi, lsb; steganografi.*

1 Pendahuluan

Latar belakang penelitian ini berkaitan dengan kerentanan penyalahgunaan informasi oleh pihak yang tidak berwenang atau seseorang yang tidak memiliki izin akses terhadap informasi yang ada. Adanya penyimpanan data atau informasi tanpa pengamanan atau membiarkan begitu saja tanpa adanya password atau enkripsi menjadi salah satu celah pihak tertentu untuk memanfaatkan kesempatan tersebut. Informasi dapat dimodifikasi, dihapus atau yang lainnya. Dalam hal ini keamanan informasi sangat diperlukan untuk menjaga kerahasiaannya dari pihak yang tidak memiliki kewenangan untuk mengaksesnya. Salah satu upaya

Diterima 04 Mei 2023 Direvisi 23 Mei 2023 Diterima untuk publikasi 31 Mei 2023

pengamanan informasi adalah pengamanan melalui enkripsi dengan algoritma kriptografi yaitu mengubah pesan asli atau informasi yang kita ingin amankan (*plaintext*) menjadi pesan acak atau informasi yang susunan kalimatnya sulit dipahami (*chipertext*) dengan menggunakan kunci (*key*)[1][2]. Pihak yang tidak berwenang akan kesulitan mengetahui maksud maupun isi dari dokumen tersebut. Apabila informasi acak tersebut (*chipertext*) ingin diubah menjadi pesan aslinya (*plaintext*) maka dilakukan proses dekripsi dengan menggunakan kunci (*key*).

Ada berbagai algoritma kriptografi saat ini, antara lain DES, RSA, AES, Blowfish, Twofish dan sebagainya[3]. Algoritma-algoritma tersebut dibedakan dalam penggunaan kunci (*key*) dalam proses enkripsinya yaitu kunci simetrik dan kunci asimetrik[4][5]. Kunci simetrik yaitu kunci yang digunakan dalam proses enkripsi maupun dekripsi dengan kunci yang sama. Begitu sebaliknya, kunci asimetrik yaitu kunci yang digunakan dalam proses enkripsi maupun dekripsi dengan kunci yang berbeda. Dalam penelitian ini penulis mengunakan algoritma Blowfish karena hasil penelitian tentang evaluasi komprehensif antara kriptografi DES, 3DES, AES, RSA dan Blowfish, Blowfish mengkonsumsi memori lebih sedikit dan RSA mengunakan memori paling besar. AES dan DES mengkonsumsi memori sedang. Dalam hal kecepatan untuk proses enkripsi dan dekripsi, Blowfish yang paling cepat diantara metode lainnya[3]. Berbanding terbalik dengan RSA yang paling lambat dalam hal kecepatan.

Penulis melakukan upaya pengamanan ganda dalam penelitian ini, karena pesan acak atau *chipertext* hasil proses enkripsi kriptografi akan menimbulkan kecurigaan oleh orang lain secara kasat mata. Pengamanan ganda tersebut adalah dengan menyisipkan *chipertext* ke dalam citra digital karena citra digital yang telah disisipi *chipertext* secara kasat mata atau visual mata manusia tidak ada perubahan, baik sebelum atau sesudah disisipkan pesan. Algoritma yang digunakan dalam steganografi ini adalah metode LSB (*least significant bit*). Metode LSB adalah menyisipkan (*embedded*) pesan rahasia ke dalam citra digital dengan cara mengganti bit ke 8, 16 dan 24 pada representasi biner di file citra digital. Maka setiap pixel file BMP 24 bit, dapat disisipkan dengan 3 bit pesan[6][7].

Dalam penelitian ini, penulis juga menggunakan kompresi huffman sebagai metode kompresi file sebelum dilakukan proses enkripsi, agar ukuran file sebelum dienkripsi menjadi lebih kecil, sehingga proses enkripsi menjadi lebih cepat [8] [9][10][11]. Berikut penelitian terdahulu yang sejenis mengenai keamanan dokumen :

1. Penelitian yang dilakukan oleh Yaya Sudarya Triana dan Astari Retnowardhani pada tahun 2018 dengan judul "*Blowfish Algorithm and Huffman Compression for Data Security Application*"[2]. Penelitian ini bertujuan untuk

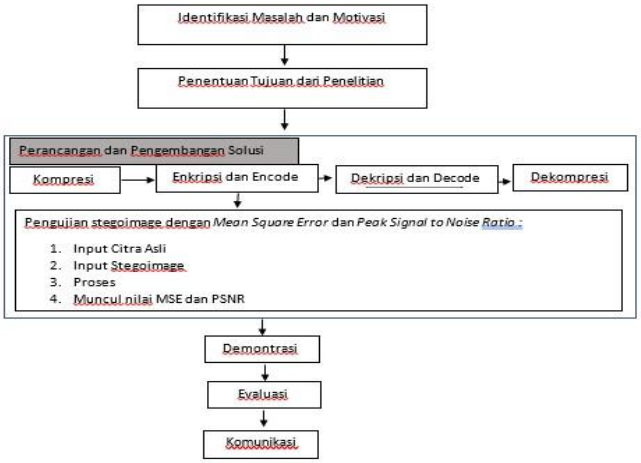
merancang aplikasi keamanan data atau dokumen dengan kriptografi Blowfish dan kompresi Huffman di rumah sakit.

2. Penelitian dengan judul "Aplikasi Enkripsi Kriptografi dengan Algoritma Blowfish dan Kompresi Huffman dalam Security Dokumen" oleh Siti Muryanah dan Syahriani Syam pada tahun 2021 [8]. Fokus dari penelitian ini adalah perancangan aplikasi untuk mengamankan dokumen dengan kriptografi Blowfish dan kompresi Huffman dengan berbagai format file seperti .txt, .pptx, .xlsx, .docx dan .pdf.

Perbedaan penelitian yang dilakukan penulis dengan penelitian diatas adalah penulis melakukan penambahan pengamanan dokumen steganografi selain dengan kriptografi dan kompresi Huffman. Hasil enkripsi dalam kriptografi, informasi akan disisipkan ke dalam citra digital. Adapun algoritma steganografi yang digunakan adalah algoritma LSB (*least significant bit*), dimana *stegoimage* tersebut secara kasat mata, sebelum dan sesudah proses encode tidak ada perubahan.

2 Metode

Penelitian ini mengunakan metode penelitian *Design Science Research Methodology* [13], dimana metode penelitian ini terdiri dari identifikasi masalah dan motivasi, penentuan tujuan dari penelitian, perancangan dan pengembangan solusi, demonstrasi, evaluasi, dan komunikasi. Lebih detailnya mengenai metode penelitian ini, dapat dilihat dalam Gambar 1. :



Gambar 1. Metode *Design Science Research Methodology* [13]

2.1 Tahap Identifikasi Masalah dan Motivasi

Identifikasi masalah berdasarkan studi literatur atau studi pustaka yang sudah dilakukan yaitu perlunya keamanan informasi terkait akses data yang semakin mudah oleh orang yang tidak berwenang dengan teknologi yang terus berkembang pesat saat ini.

2.2 Tahap Penentuan Tujuan Penelitian

Penentuan tujuan penelitian ini berdasarkan identifikasi masalah diatas, sehingga tujuan penelitian ini adalah mengimplementasikan *hidden text* pada citra digital dengan kriptografi dan kompresi Huffman sebagai upaya dalam mengamankan informasi atau dokumen yang kita miliki.

2.3 Tahap Perancangan dan Pengembangan Solusi

Cakupan dalam tahap ini adalah perancangan sistem yang dimulai dengan proses kompresi, enkripsi dan encode, decode dan dekripsi serta dekompresi. Pengujian dalam penelitian ini mengunakan pengujian perhitungan nilai PSNR (*Peak Signal to Noise Ratio*) dan MSE (*Mean Square Error*). Perhitungan MSE ini untuk mengetahui tingkat kesalahan piksel - piksel citra digital hasil encode *(stegoimage*) terhadap citra asli (media cover). Semakin kecil nilai MSE yang dihasilkan, maka kualitas citra keluaran akan semakin baik atau dapat dikatakan semakin mendekati citra aslinya. Sedangkan perhitungan PSNR dilakukan untuk mengetahui perbandingan antara nilai maksimum dari sinyal yang diukur dengan besarnya derau yang berpengaruh pada sinyal tersebut[12].

2.4 Tahap Demonstrasi, Evaluasi dan Komunikasi

Tahapan ini bertujuan untuk mengecek kesesuaian rancangan dengan tujuan dari penelitian yang sudah ditentukan, dievalusi dan dikomunikasikan hasil penelitian yang dihasilkan.

3 Hasil dan Pembahasan

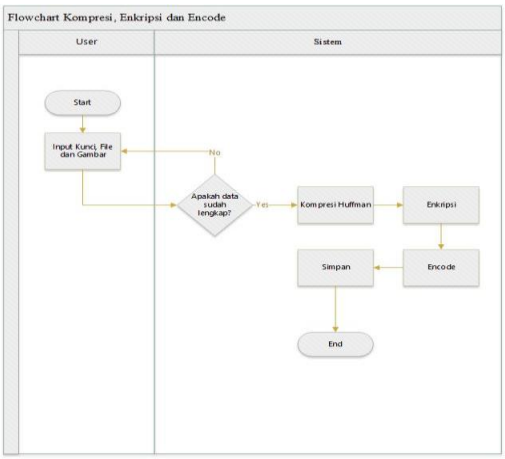
Dalam bab ini akan dibahas mengenai proses kerja sistem, hasil implementasi, tampilan antar muka dan pengujian.

3.1 Proses Kerja Sistem

Proses pengamanan dokumen dalam penelitian ini meliputi 2 proses yaitu proses kompresi, enkripsi dan *encode*, dan kedua dekripsi, *decode*, dan dekompresi.

3.1.1 Proses Kompresi, Enkripsi dan *Encode*

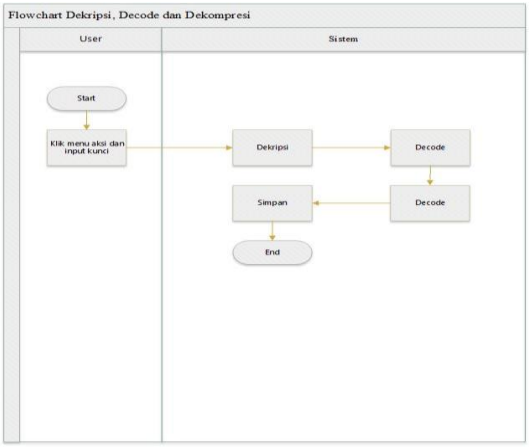
Tahap ini diawali dengan input *key* atau kunci, selanjutnya *input* file atau dokumen dan *input* citra digital dengan masing-masing ukuran maksimal 2 MB. Adapun untuk flowchart dari alur proses kompresi, enkripsi dan *encode* dipaparkan dalam gambar 2 :



Gambar 2. Flowchart Proses Kompresi, Enkripsi dan Encode

3.1.2 Dekripsi, *Decode* dan Dekompresi

Proses dekripsi, *decode* dan dekompresi dapat dilakukan melalui list hasil dekripsi dengan memilih menu aksi dan *input* kunci. Flowchart dari alur proses dekripsi, *decode* dan dekompresi dipaparkan dalam gambar 3 :



Gambar 3. Flowchart Dekripsi, *Decode* dan Dekompresi

3.2 Hasil Implementasi

Implementasi dari sistem ini dilakukan sebanyak 15 kali pengamanan file dengan beberapa tipe file dan citra digital, yaitu dengan format pdf, xlsx, docx, ppt dan txt dengan masing-masing 3 file dan 3 cover agar hasil yang diperoleh lebih valid dan maksimal. Hasil yang diperoleh adalah proses kompresi, enkripsi, encode dan dekripsi, decode dan dekompresi berhasil dilakukan dengan baik, dengan kata lain bahwa proses pengamanan file atau dokumen berhasil atau sukses. Masing-masing file dapat dikompresi dengan baik dengan perubahan ukuran file sebelum dan sesudah enkripsi yang cukup signifikan. Waktu proses masing-masing file juga cukup singkat sekitar 1 sampai 1-8 detik saja. Detail implementasi dari penelitian ini dipaparkan dalam tabel 1 :

Tabel 1. Hasil Implementasi

No	File	Cover	Inputan Enkripsi & Encode		Durasi Enkripsi	Output Enkripsi & Encode		Output Dekripsi & Decode		Durasi Dekripsi
			Size File	Size Gambar		Size File	Size Gambar	Size File	Size Gambar	
1	pdf1	jpg1	31.37 KB	106.16 KB	00:00:08	20.47 KB	3.52 MB	31.37 KB	106.16 KB	00:00:00
2	pdf2	jpg2	134.41 KB	87.01 KB	00:00:06	34.00 KB	974.07 KB	134.41 KB	87.01 KB	00:00:00
3	pdf3	jpg3	138.91 KB	31.02 KB	00:00:01	32.16 KB	117.65 KB	138.91 KB	31.02 KB	00:00:00
4	xlsx1	jpg1	11.57 KB	34.34 KB	00:00:03	7.38 KB	117.65 KB	11.57 KB	34.34 KB	00:00:00
5	xlsx2	jpg2	11.72 KB	5.36 KB	00:00:04	6.66 KB	147.93 KB	11.72 KB	5.36 KB	00:00:00
6	xlsx3	jpg3	10.36 KB	14.85 KB	00:00:00	6.92 KB	117.65 KB	10.36 KB	14.85 KB	00:00:00
7	docx1	jpg1	103.48 KB	14.27 KB	00:00:02	14.71 KB	117.65 KB	103.48 KB	14.27 KB	00:00:00
8	docx2	jpg2	82.94 KB	34.34 KB	00:00:01	9.62 KB	117.65 KB	82.94 KB	34.34 KB	00:00:00
9	docx3	jpg3	212.89 KB	14.27 KB	00:00:02	29.54 KB	117.65 KB	212.89 KB	14.27 KB	00:00:00
10	pptx1	jpg1	655.17 KB	87.01 KB	00:00:05	19.51 KB	974.07 KB	655.17 KB	87.01 KB	00:00:00
11	pptx2	jpg2	950.22 KB	106.16 KB	00:00:01	13.85 KB	3.52 MB	950.22 KB	106.16 KB	00:00:01
12	pptx3	jpg3	759.81 KB	13.56 KB	00:00:01	17.74 KB	133.25 KB	759.81 KB	13.56 KB	00:00:00
13	txt1	jpg1	1.52 KB	31.02 KB	00:00:00	1.52 KB	117.65 KB	1.52 KB	31.02 KB	00:00:00
14	txt2	jpg2	168 Byte	97.38 KB	00:00:01	168 Byte	980.51 KB	168 Byte	97.38 KB	00:00:00
15	txt3	jpg3	2.21 KB	3.28 KB	00:00:00	2.21 KB	38.51 KB	2.21 KB	3.28 KB	00:00:00

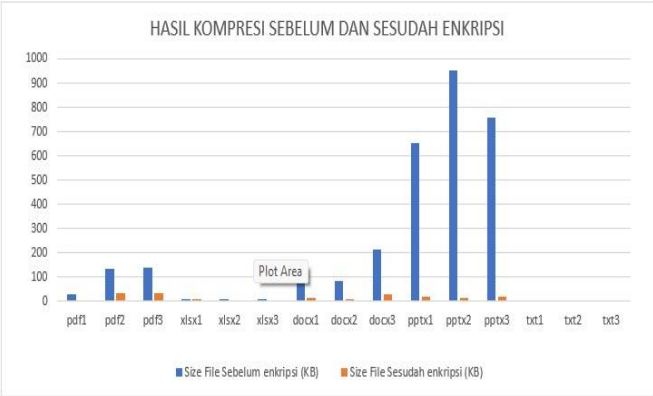
Diterima 04 Mei 2023 Direvisi 23 Mei 2023 Diterima untuk publikasi 31 Mei 2023

Dari hasil implementasi diatas diperoleh bahwa proses kompresi berhasil memperkecil ukuran file asli sebelum dienkripsi. Besarnya tingkat kompresi (%) dari data diatas rata-rata sebesar 57% dengan detail masing-masing file, terdapat dalam tabel 2:

Tabel 1. Detail Tingkat Hasil Kompresi

No	File	Cover	Size File Sebelum enkripsi (KB)	Size File Sesudah enkripsi (KB)	Besarnya tingkat kompresi (%)
1	pdf1	jpg1	31,37	20,47	35%
2	pdf2	jpg2	134,41	34	75%
3	pdf3	jpg3	138,91	32,16	77%
4	xlsx1	jpg1	11,57	7,38	36%
5	xlsx2	jpg2	11,72	6,66	43%
6	xlsx3	jpg3	10,36	6,92	33%
7	docx1	jpg1	103,48	14,71	86%
8	docx2	jpg2	82,94	9,62	88%
9	docx3	jpg3	212,89	29,54	86%
10	pptx1	jpg1	655,17	19,51	97%
11	pptx2	jpg2	950,22	13,85	99%
12	pptx3	jpg3	759,81	17,74	98%
13	txt1	jpg1	1,52	1,52	0%
14	txt2	jpg2	0,1640625	0,1640625	0%
15	txt3	jpg3	2,21	2,21	0%
					57%

Adapun untuk grafik dari tingkat hasil kompresi dapat dijelaskan dalam gambar 4 :



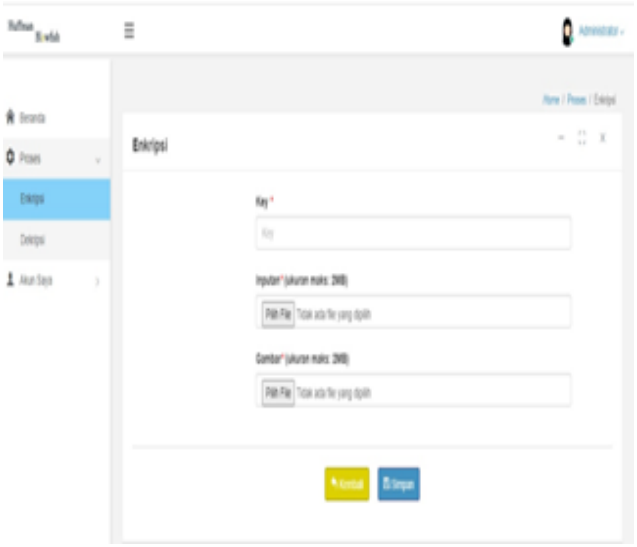
Gambar 4. Grafik Hasil Kompresi Sebelum dan Sesudah Enkripsi

Dari gambar 4 diatas dapat menjelaskan bahwa kompresi Huffman bisa memperkecil ukuran file yang signifikan antara sebelum dikompresi dan sesudah dikompresi. File yang memiliki format pptx (file yang tersimpan di power point) memiliki perbedaan yang cukup jauh antara sebelum dan sesudah dikompresi, dari 655,17 KB menjadi 87.01 KB dengan tingkat prosentase perubahan ukuran 97%, dari 950,22 KB menjadi 13,85 KB dengan tingkat prosentase perubahan 99% dan dari 759,81 KB menjadi 17,74 KB dengan tingkat prosentase perubahan 98%.

3.3 Tampilan Antar Muka

Tampilan antar muka dalam proses kompresi, enkripsi dan encode sebagai berikut :

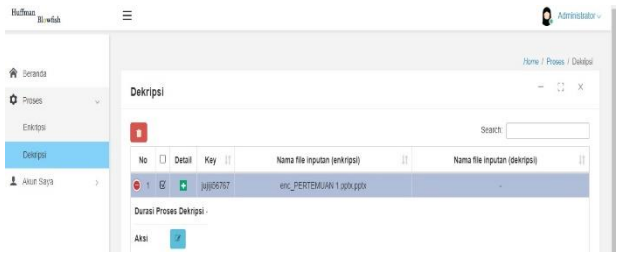
3.3.1 Antar Muka Proses Enkripsi dan *Encode*



Gambar 5. Tampilan Antar muka Proses Enkripsi dan Encode

Gambar 5 menggambarkan bahwa untuk melakukan proses enkripsi dengan *input* kunci lalu *input* file atau dokumen dalam format txt, docx, xlsx, pdf dan ppt dan gambar atau citra digital. Adapun ukuran dokumen atau citra digital maksimal 2 MB. Klik simpan.

3.3.2 Antar Muka Proses Dekripsi dan *Decode*



Gambar 6. Tampilan Antar Muka Proses Dekripsi dan *Decode*

Gambar 6 menjelaskan bahwa proses dekripsi dan *decode* dapat dilakukan dengan klik tanda plus di nomor urut baris yang akan di dekripsi, lalu klik menu aksi. Baris field tersebut merupakan field hasil enkripsi sebelumnya. Setelah itu akan muncul kunci dan klik ubah.

3.4 Pengujian Perhitungan nilai MSE dan PSNR

Menurut [14] *Peak Signal to Noise Ratio* adalah perbandingan antara nilai maksimum dari sinyal yang diukur dengan besarnya derau yang berpengaruh pada sinyal tersebut. PSNR biasanya diukur dalam satuan desibel (dB). Nilai PSNR jatuh dibawah 30 dB mengindikasikan kualitas yang relatif rendah. Kualitas *stegoimage* yang tinggi berada pada nilai 40dB dan diatasnya [6][15]. Pengujian nilai PSNR digunakan untuk mengetahui perbandingan kualitas citra sebelum dan sesudah disisipkan pesan. Untuk menentukan nilai PSNR dapat dikalkulasikan dengan persamaan berikut[6] :

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right)$$
$$= 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right)$$
$$= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE)$$

……..(1)

dimana:
PSNR adalah *Peak Signal to Noise Ratio*.
MAX adalah nilai maksimum dari piksel citra.
MSE adalah nilai *Mean Square Error* dari citra.

Mean Square Error (MSE) merupakan parameter yang menunjukkan tingkat kesalahan piksel - piksel citra hasil pemrosesan sinyal (*stegoimage*) terhadap citra asli (media cover). Semakin kecil nilai MSE yang dihasilkan maka semakin kualitas citra keluaran akan semakin baik atau dapat dikatakan semakin mendekati citra aslinya. Nilai MSE dapat dikalkulasi dengan persamaan sebagai berikut [6]:

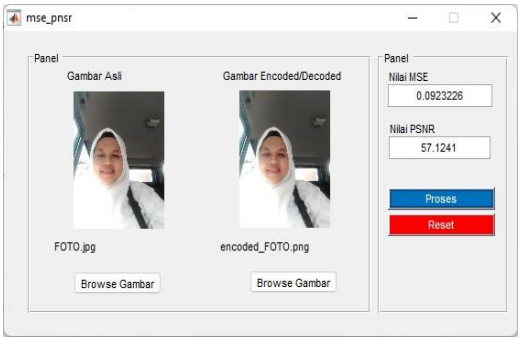
$$MSE = \frac{1}{m \times n}\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}[f(i,j) - g(i,j)]^2$$

…….. (2)

dimana:
MSE adalah nilai *Mean Square Error* dari citra.
m adalah panjang citra *stego* (dalam piksel).
n adalah ukuran width citra *stego* (dalam piksel).
f(i,j) adalah nilai piksel dari citra cover.
g(i,j) adalah nilai piksel pada citra *stego*

Dalam pengujian sistem ini, *stegoimage* yang akan diuji masing-masing hasil dari proses encode dengan dari penyisipan file dengan format txt, pdf, pptx, docx, xlsx. Masing-masing format file tersebut akan diuji sebanyak 3 kali dengan file

yang berbeda tapi format file yang sama untuk memastikan tingkat validasi hasil pengujian. Gambar 7 menunjukkan penggujian perhitungan nilai MSE dan PSNR.



Gambar 7. Pengujian Perhitungan nilai MSE dan PSNR

Tabel 3 merupakan hasil pengujian *stegoimage* sebelum dan sesudah dilakukan *encode* dengan pengujian MSE dan PSNR :

Tabel 2. Hasil Pengujian Stegoimage dengan MSE dan PSNR

No	File	Cover	PENGUJIAN	
			MSE	PSNR
1	pdf1	jpg1	0,092	57,12
2	pdf2	jpg2	2,6294	45,5353
3	pdf3	jpg3	5,89837	38,1569
4	xlsx1	jpg1	2,88105	43,1446
5	xlsx2	jpg2	2,72906	45,771
6	xlsx3	jpg3	2,50955	46,0471
7	docx1	jpg1	3,15508	41,583
8	docx2	jpg2	2,90948	43,0691
9	docx3	jpg3	5,80225	37,5244
10	pptx1	jpg1	2,56682	46,0284
11	pptx2	jpg2	0,122414	55,4124
12	pptx3	jpg3	3,39445	39,2532
13	txt1	jpg1	2,5708	45,7204
14	txt2	jpg2	2,84966	43,8774
15	txt3	jpg3	2,41881	39,8809
			0,092	**57,12**

Tabel 3 diatas menjelaskan bahwa hasil proses *encode* menghasilkan *stegoimage* yang berkualitas tinggi yaitu dengan nilai MSE yang cukup rendah rata-rata 0,092 dan nilai PSNR 57,12 dB

4 Kesimpulan

Kesimpulan penelitian ini adalah sistem yang dibangun dapat mengamankan file dokumen dalam format pdf, xlsx, docx, pptx dan txt dengan melakukan proses enkripsi, *encode*, dekripsi dan *decode*. Ukuran file menjadi lebih kecil dengan tingkat prosentase perubahan rata-rata sebesar 57% dari ukuran file aslinya (kompres). Pengujian dengan perhitungan nilai MSE dan PSNR menghasilkan *stegoimage* dengan kualitas tingkat tinggi yaitu nilai MSE sangat kecil, rata-rata 0,092 dan nilai PSNR rata-rata 57,12 dB.

5 Referensi

[1] T. S. Barhoom, S. Mohammed, and A. Mousa, "A Steganography LSB Technique for Hiding Image Within Image Using Blowfish Encryption Algorithm," *Int. J. Res. Eng. Sci.*, vol. 3, no. 31, pp. 61–66, 2015, [Online]. Available: www.ijres.org.

[2] Y. S. Triana and A. Retnowardhani, "Blowfish algorithm and Huffman compression for data security application," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 453, no. 1, 2018, doi: 10.1088/1757-899X/453/1/012074.

[3] P. Princy, "A Comparison Of Symmetric Key Algorithms Des , Aes , Blowfish , RC4, RC6 : A SURVEY," vol. 6, no. 05, pp. 328–331, 2015.

[4] A. Rana, "A Symmetrical key Cryptography Analysis using Blowfish Algorithm A Symmetrical key Cryptography Analysis using Blowfish Algorithm," no. July, 2016, doi: 10.17577/IJERTV5IS070276.

[5] "A Comparative Study and Performance Evaluation of Cryptographic Algorithms : AES and Blowfish," no. December, 2016.

[6] A. H. Zyara, "Suggested Method for Encryption and Hiding Image using LCG and LSB," no. 26, pp. 66–77, 2018.

[7] A. D. V Math, M. Damrudi, and K. J. Aval, "Image Steganography on Compressed and Encrypted Message," vol. 9, no. 9, pp. 7321–7330, 2020.

[8] S. Muryanah and S. Syam, "Aplikasi Enkripsi Kriptografi dengan Algoritma Blowfish dan Kompresi Huffman dalam Security Dokumen," vol. 10, no. 02, pp. 24–34, 2021.

[9] A. S. Wahyu Pramusinto, Nugroho Wizaksono, "Aplikasi Pengamanan File Dengan Metode Kriptografi AES 192, RC4 Dan Metode Kompresi Huffman,"

vol. 16, no. 2, pp. 47–53, 2019.

[10] A. Pahdi, "Algoritma Huffman Dalam Pemampatan Dan Enksripsi Data," *IJNS - Indones. J. Netw. Secur.*, vol. 6, no. 3, pp. 1–7, 2017, [Online]. Available: http://ijns.org/journal/index.php/ijns/article/view/1461.

[11] E. Prayoga and K. M. Suryaningrum, "Implementasi Algoritma Huffman Dan Run Length Encoding Pada Aplikasi Kompresi Berbasis Web," *J. Ilm. Teknol. Infomasi Terap.*, vol. 4, no. 2, pp. 92–101, 2018, doi: 10.33197/jitter.vol4.iss2.2018.154.

[12] H. A. Pradana, D. Y. Sylfania, and F. P. Juniawan, "Perbandingan kinerja RSA dan AES terhadap kompresi pesan SMS menggunakan algoritme Huffman Performance comparison of RSA and AES to SMS messages compression using," vol. 8, no. July, pp. 171–177, 2020, doi: 10.14710/jtsiskom.2020.13468.

[13] S. Ken Peffers, Tuure Tuunanen, Marcus A. Rothenberger Chatterjee, "A Design Science Research Methodology for Information Systems Research," vol. 24, no. 3, pp. 45–78, 2007.

[14] J. Moreno, B. Jaime, and S. Saucedo, "Towards No-Reference of Peak Signal to Noise Ratio Estimation Based on Chromatic Induction Model," vol. 4, no. 1, pp. 123–130, 2013.

[15] R. Bhardwaj and V. Sharma, "Image Steganography Based on Complemented Message and Inverted bit LSB," *Procedia - Procedia Comput. Sci.*, vol. 93, no. September, pp. 832–838, 2016, doi: 10.1016/j.procs.2016.07.245.