APLIKASI KEAMANAN PESAN (.TXT) MENGGUNAKAN METODE *TRIPLE* DES DAN METODE KOMBINASI LSB DAN BLUM-BLUM-SHUB

Derwin R Sina[1], Guido Alfino Kiu[2*], Bertha S Djahi[3] dan Emerensye S Y Pandie[4]

[1,2,3,4]Program Studi Ilmu Komputer, Universitas Nusa Cendana, Jl. Adisucipto Penfui Kupang, Indonesia
[1]Email: derwinsina@staf.undana.ac.id
[2]Email: guidoakiu@gmail.com
[3]Email: bertha.djahi@staf.undana.ac.id
[4]Email: emerensyepandie@staf.undana.ac.id

ABSTRAK

Keamanan merupakan salah satu aspek penting dalam proses pertukaran informasi (pesan). Untuk mencegah penyalahgunaan atau serangan oleh pihak yang tidak berwenang (*attacker*) terhadap pesan yang bersifat privasi, maka pesan tersebut harus diamankan. Ada beberapa metode yang dapat digunakan dalam mengamankan pesan, salah satunya adalah dengan mengombinasikan metode kriptografi *triple data encryption standard* (DES), pembangkit bilangan acak Blum-Blum-Shub (BBS), dan steganografi *least significant bit* (LSB). Pada penelitian ini, metode kriptografi *Triple* DES digunakan untuk mengenkripsi pesan (*embedded-message*) dengan ekstensi .txt dan metode pembangkit bilangan acak BBS digunakan untuk menentukan posisi *pixel* secara acak yang akan disisipi pesan pada *cover-image*. Metode steganografi LSB digunakan untuk melakukan proses penyisipan (*embedding*) terhadap *embedded-message* yang telah terenkripsi pada posisi *pixel* yang dihasilkan dari proses pembangkitan bilangan acak BBS. Hasil pengujian menunjukkan bahwa sistem dapat mengekstraksi *embedded message* yang disembunyikan di dalam sebuah *stego-image* dengan tingkat akurasi 100%. Jumlah maksimal karakter *embedded-message* yang dapat digunakan dalam pengujian adalah sebanyak 150 karakter. Pengujian juga menghasilkan *stego-image* yang memiliki rata-rata nilai *peak signal to noise ratio* (PSNR) sama dengan 88,61 yang berarti bahwa *stego-image* yang dihasilkan memiliki kualitas tinggi (tidak terjadi penurunan mutu secara signifikan) dan keberadaan pesan di dalam *stego-image* semakin sulit untuk diketahui (*imperceptibility*).
Kata kunci: Kriptografi, Steganografi, DES, LSB, BBS

ABSTRACT

Security is one of the important aspects of the process of exchanging information (messages). To prevent misuse or attacks by unauthorized parties (attackers) on private messages, the message must be secured. Several methods can be used to secure messages, one of which is by combining the triple data encryption standard (DES) cryptography method, Blum-Blum Shub (BBS) random number generator, and least significant bit (LSB) steganography. In this study, the triple DES cryptographic method is used to encrypt messages (embedded-message) with the extension .txt and the BBS random number generator method is used to determine the position of a random pixel to be inserted in the cover-image message. The LSB steganography method is used to perform the embedding process of the encrypted embedded-message at the pixel position resulting from the BBS random number generation process. The test results show that the system can extract embedded messages hidden in a stego-image with 100% accuracy. The maximum number of embedded-message characters that can be used in the test is 150 characters. The test also produces a stego-image that has an average peak signal to noise ratio (PSNR) value of 88.61, which means that the resulting stego-image has high quality (no significant quality degradation) and the presence of messages in the stego-image is getting harder to detect (imperceptibility).
Keywords: Cryptography, Steganography, DES, LSB, BBS

1. PENDAHULUAN

Keamanan merupakan salah satu aspek penting dalam proses pertukaran informasi. Untuk mencegah penyalahgunaan atau serangan oleh pihak yang tidak berwenang (*attacker*) terhadap pesan yang bersifat rahasia, maka pesan tersebut harus diamankan.

Steganografi merupakan metode yang dapat digunakan dalam mengamankan pesan [1]. Salah satu metode steganografi adalah *least significant bit* (LSB). LSB bekerja dengan cara mengganti bit *pixel* yang

ISSN: 2337-7631 (printed)
ISSN: 2654-4091 (online)

memiliki nilai tidak berarti (*least significant bit*) pada *cover-object* dengan bit *embedded message* secara berurutan [2]. Untuk meningkatkan keamanan, maka pesan tidak disisipi pada *pixel* secara secara berurutan, tetapi disisipkan pada *pixel* yang dipilih secara acak [3]. Pembangkitan *pixel* secara acak dapat menggunakan metode Blum-Blum-Shub (BBS). Kelebihan dari BBS adalah metode yang digunakan sangat sederhana dan paling mangkus secara kompleksitas teoritis [4]. Dalam penerapannya, proses *embedding* pada steganografi dapat dikombinasikan dengan metode kriptografi [5]–[13]. Proses enkripsi dengan menggunakan metode kriptografi akan dilakukan sebelum proses *embedding* dengan metode steganografi. Salah satu metode kriptografi adalah *triple data encryption standard* (DES). Kelebihan dari triple DES adalah metode yang digunakan lebih mangkus dibandingkan metode DES dan *double* DES.

2. MATERI DAN METODE

Embedding

Embedding adalah proses menyisipkan *embedded message* ke dalam sebuah *cover-object*. Proses ini menerima masukan berupa beberapa parameter dari *user* seperti *embedded message*. Setelah *embedded message* dimasukkan oleh *user*, proses *embedding* dapat dilanjutkan dengan tahap-tahap sebagai berikut:

1. Tahap pembangkitan bilangan acak. Metode yang digunakan dalam tahap ini adalah metode BBS. Algoritma untuk membangkitkan bilangan acak dengan BBS adalah sebagai berikut:
   a. Pilih 2 buah bilangan prima yang akan dijadikan sebagai *key*, *p* dan *q*, yang masing-masing kongruen (≡) dengan 3 modulo 4.
   b. Hitung nilai n (bilangan bulat Blum) dengan menggunakan persamaan 1.
   $$n = p \times q \quad (1)$$
   c. Pilih bilangan bulat acak lain, s, sebagai umpan sedemikian sehingga:
      i. $2 \leq s \leq n$
      ii. *s* dan *n* relatif prima. *s* dan *n* dikatakan relatif prima apabila faktor persekutuan terbesar dari *s* dan *n* sama dengan 1 (FPB (*s*, *n*) = 1). Kemudian hitung x_0 dengan menggunakan persamaan 2.
      $$x_0 = s^2 \ mod \ n \quad (2)$$
   d. Barisan bit acak dihasilkan dengan melakukan iterasi berikut sepanjang yang diinginkan:
      i. Hitung x_i dengan menggunakan persamaan 3.
      $$x_i = x_{i-1} 2 \ mod \ n \quad (3)$$
      ii. z_i = bit LSB dari x_i
      Barisan bit acak adalah, z_1, z_2, z_3, ...

   Sebelum melakukan pembangkitkan bilangan acak, terlebih dahulu sistem akan menghitung jumlah bilangan acak yang dibutuhkan (BAD) dengan menggunakan persamaan 4 atau 5. Jika jumlah karakter *embedded message* (JKEM) merupakan kelipatan 8 maka
   $$BAD = 8 \times JKEM \quad (4)$$
   Jika JKEM bukan kelipatan 8 maka
   $$AD = 8 \times (JKEM + 8 - (JKEM \ mod \ 8)) \quad (5)$$
   Misalkan, *embedded message* adalah "COMPUTER" maka nilai BAD dapat dihitung dengan menggunakan persamaan 4 sehingga menghasilkan nilai 64. Setelah itu, sistem akan membangkitkan pilihan parameter BBS (*p*-*q*-*s*) dan ukuran minimal *cover-image* (UMC) dari Tabel 1 sesuai dengan nilai BAD. Selanjutnya, *user* dapat memilih parameter BBS yang diinginkan. Terakhir, sistem akan membangkitkan bilangan acak sebanyak nilai BAD dengan metode BBS. Bilangan acak yang dibangkitkan merupakan nilai yang merepresentasikan posisi *pixel* pada *cover-image* yang akan disisipi *embedded message*.
2. Tahap enkripsi**.** Tahap enkripsi dengan menggunakan metode triple DES dilakukan sebanyak 3 putaran DES dengan kunci enkripsi yang berbeda dari *user*. Untuk setiap putaran, enkripsi terhadap *plaintext* (*embedded message*) dibagi menjadi 3 tahap yaitu permutasi awal, *enciphering*, dan permutasi akhir. *Ciphertext* hasil dari putaran ketiga enkripsi merupakan *ciphertext* yang akan disisipi pada *cover-image*.
3. Tahap *embedding*. Setelah tahap pembangkitan bilangan acak dan enkripsi selesai, proses selanjutnya adalah melakukan *embedding* dengan menggunakan metode LSB. Sebelum melakukan proses *embedding*, terlebih dahulu *user* harus memasukkan *cover-image* dengan jumlah *pixel* minimal yaitu sama dengan UMC dan komponen warna yang akan disisipi (R/G/B). Setelah *cover-image* yang dimasukkan oleh *user* memiliki jumlah *pixel* yang lebih besar atau sama dengan UMC, maka posisi *pixel* (*x* dan *y*) yang akan menjadi tempat disisipkannya bit *ciphertext* dapat dihitung dengan menggunakan bilangan acak BBS yang telah dihasilkan, persamaan 6 dan 7. Setelah itu, *embedding* dapat dilakukan dengan menggunakan persamaan 8.

ISSN: 2337-7631 (printed)
ISSN: 2654-4091 (online)

$$y = Floor\ (Posisi\ \div Width) \quad (6)$$

$$x = Posisi - (y\ \times Width) \quad (7)$$

$$Pixel = 2\ \times Floor\left(\frac{Pixel}{2}\right) + Bit\ Embedded\ Message\ (0\ atau\ 1) \quad (8)$$

Tabel 1. Hubungan antara parameter BBS, JKEM, BAD, dan UMC

p	*q*	*s*	Jumlah Bilangan Acak yang dapat dibangkitkan	JKEM	BAD	Bilangan acak terbesar
499	7	2	81	8	64	3.481
103	107	2	103	8	64	10.956
103	163	2	215	8	64	16.688
103	179	2	87	8	64	18.384
103	223	2	70	8	64	22.092
571	199	2	89	8	64	113.476
251	107	2	259	25	256	26.622
163	199	2	269	25	256	31.842
263	151	2	258	25	256	39.578
179	223	2	395	25	256	39.504
103	419	2	359	25	256	43.014
263	103	2	518	50	448	27.049
127	347	2	515	50	448	43.958
199	107	2	779	50	448	21.278
547	167	2	491	50	448	90.862
499	211	2	491	50	448	104.782
499	239	2	983	100	832	118.788
599	167	2	1201	100	832	99.915
499	107	2	1.205	100	832	53.375
619	107	2	2.651	150	1.216	66.105
719	251	2	1.761	150	1.216	180.178
647	283	2	1.655	150	1.216	182.872
983	991	2	2.938	150	1.216	972.988

***Peak Signal to Noise Ratio* (PSNR)**

Untuk menilai kualitas dari *stego-image* yang dihasilkan bagus atau tidaknya dapat menggunakan nilai PSNR. Sebelum menghitung nilai dari PSNR, terlebih dahulu harus dicari nilai dari *root mean square* (RMS). RMS dapat dihitung dengan persamaan 9.

$$RMS = \sqrt{\frac{1}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M}(I_{ij} - \acute{I}_{ij})^2} \quad (9)$$

Setelah nilai dari *rms* diperoleh, nilai dari PSNR dapat dihitung dengan persamaan 10.

$$PSNR = 20\ \times \log_{10}\left(\frac{255}{rms}\right) \quad (10)$$

Satuan dari PSNR adalah desibel (dB). Citra dengan nilai PSNR ≥ 40 dapat dianggap memiliki kualitas yang tinggi [14].

3. HASIL DAN PEMBAHASAN

Hasil

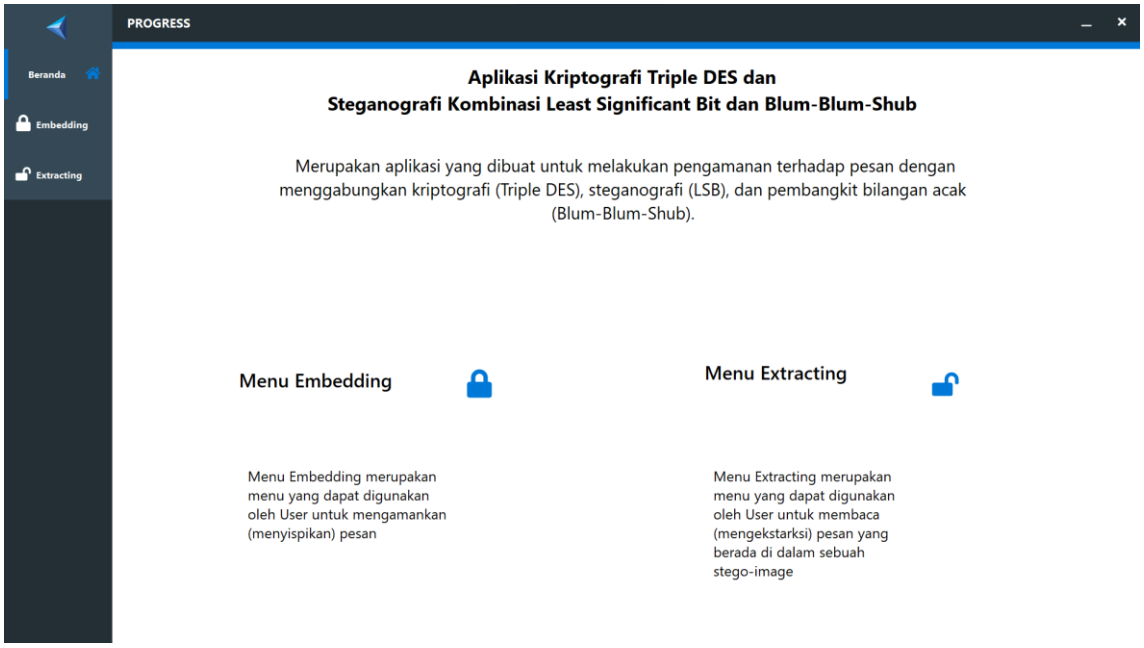
Setelah *user* memasukkan parameter yang dibutuhkan saat proses *embedding*, sistem akan menghasilkan sebuah *stego-image* yang berisi *embedded message* yang telah terenkripsi. Selain itu, *user* juga dapat memasukkan parameter yang dibutuhkan saat proses *extracting* untuk mengekstraksi *embedded message* yang disembunyikan di dalam sebuah *stego-image*. Tampilan aplikasi dapat dilihat pada gambar 1 – gambar 3.

Pembahasan

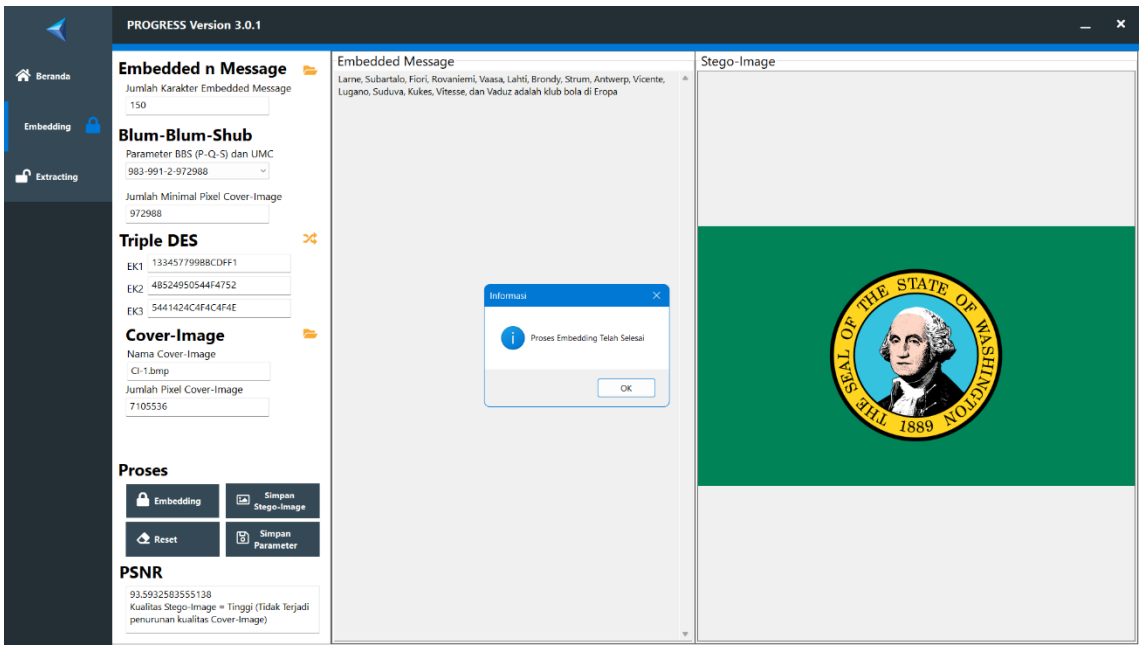
Pengujian sistem dilakukan dengan menggunakan 10 *embedded message* (.txt) dan 10 *cover-image* (.bmp). Setiap *embedded message* akan disisipi pada 2 *cover-image* secara acak sehingga jumlah percobaan yang akan dilakukan adalah 20 percobaan. Pengujian dilakukan dengan 2 cara yaitu pengujian kesesuaian antara proses *embedding* dan *extracting* dan pengujian kualitas *stego-image*. Semua *embedded message* yang dihasilkan dari proses *extracting* sesuai dengan *embedded message* yang digunakan saat proses *embedding*. Hal ini menunjukkan bahwa sistem dapat mengekstraksi *embedded message* yang disembunyikan di dalam sebuah *cover-image* dengan tingkat akurasi 100%. Pengujian kualitas *stego-*

ISSN: 2337-7631 (printed)
ISSN: 2654-4091 (online)

image menghasilkan rata-rata nilai PSNR sama dengan 88,61 (dapat dilihat pada tabel 2). Nilai rata-rata tersebut lebih besar sama dengan 40 yang berarti bahwa *stego-image* yang dihasilkan memiliki kualitas yang tinggi (tidak terjadi penurunan mutu secara signifikan) [14] dan keberadaan pesan di dalam *stego-image* semakin sulit untuk diketahui oleh pihak lain (*imperceptibility*). Berdasarkan hasil pengujian ini juga, dapat disimpulkan pula bahwa terdapat hubungan JKEM terhadap nilai PSNR. Keterkaitan tersebut dapat dilihat pada tabel 2.

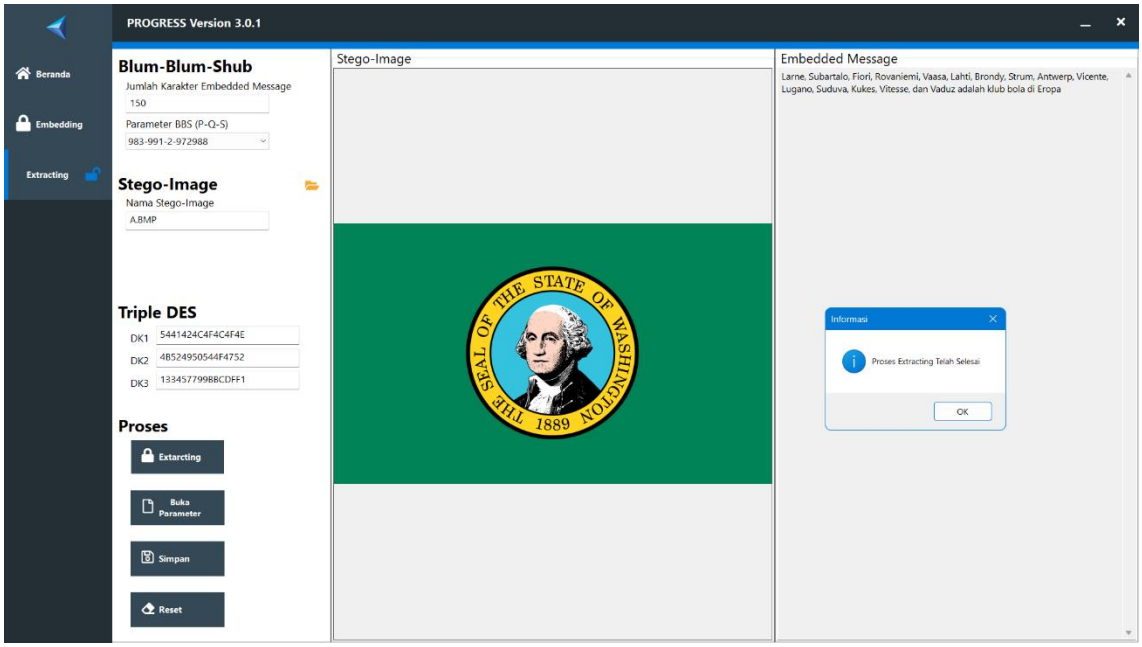


Gambar 1. Tampilan Menu Beranda



Gambar 2. Tampilan Menu *Embedding*

ISSN: 2337-7631 (printed)
ISSN: 2654-4091 (online)



Gambar 3. Tampilan Menu *Extracting*

Tabel 2. Hubungan JKEM terhadap Nilai PSNR

Cover-Image	Nomor Percobaan	*Stego-Image*	JKEM	PSNR
CI-1	8	SI-8	25	95,75
	20	SI-20	150	89,01
CI-2	16	SI-16	100	89,45
	19	SI-19	150	87,93
CI-3	4	SI-4	8	98,51
	17	SI-17	150	86,39
CI-4	11	SI-11	50	85,80
	15	SI-15	100	83,53
CI-5	6	SI-6	25	83,33
	18	SI-18	150	77,39
CI-6	3	SI-3	8	87,30
	7	SI-7	25	81,31
CI-7	2	SI-2	8	86,88
	5	SI-5	25	81,67
CI-8	12	SI-12	50	92,93
	14	SI-14	100	90,32
CI-9	9	SI-9	50	92,48
	13	SI-13	100	89,66
CI-10	1	SI-1	8	100,40
	10	SI-10	50	92,13

Berdasarkan tabel 2 dapat disimpulkan bahwa semakin besar JKEM yang disisipi pada sebuah *cover-image* maka nilai PSNR akan lebih kecil dibandingkan dengan *cover-image* yang disisipi JKEM dengan jumlah yang lebih sedikit.

4. KESIMPULAN DAN SARAN

Kesimpulan

Berdasarkan penelitian yang telah dilakukan dengan menggunakan metode kriptografi *triple* DES dan metode kombinasi steganografi LSB dan BBS, dapat disimpulkan bahwa setelah dilakukan 20 kali percobaan, semua hasil dari proses *extracting* berupa *embedded message* sesuai dengan *embedded message* yang digunakan saat proses *embedding*. Hal ini menunjukkan bahwa sistem dapat mengekstraksi *embedded message* yang disembunyikan di dalam sebuah *cover-image* dengan tingkat akurasi 100%. Dua puluh *stego-image* yang dihasilkan memiliki nilai rata-rata PSNR sebesar 88,61. Nilai tersebut menunjukkan bahwa

ISSN: 2337-7631 (printed)
ISSN: 2654-4091 (online)

stego-image yang dihasilkan memiliki kualitas yang tinggi (tidak terjadi penurunan mutu secara signifikan) dan keberadaan pesan di dalam *stego-image* semakin sulit untuk diketahui oleh pihak lain (*imperceptibility*). Selain itu, berdasarkan percobaan yang telah dilakukan dapat disimpulkan pula bahwa semakin besar JKEM yang disisipi pada sebuah *cover-image* maka nilai PSNR akan lebih kecil dibandingkan dengan *cover-image* yang disisipi JKEM dengan jumlah yang lebih sedikit.

Saran

Pada penelitian ini, pesan yang diamankan dibatasi dengan maksimal 150 karakter dengan ekstensi .txt, berdasarkan hal tersebut maka peneliti menyarankan untuk penelitian selanjutnya dapat menggunakan pesan dengan jumlah karakter di atas 150 karakter dan ekstensi yang dapat digunakan adalah .pdf, .docx, .png, dan lain sebagainya.

DAFTAR PUSTAKA

[1] N. F. Johnson and S. Jajodia, "Exploring steganography: Seeing the unseen," *Computer*, vol. 31, no. 2, pp. 26–34, 1998.
[2] M. Asad, J. Gilani, and A. Khalid, "An enhanced least significant bit modification technique for audio steganography," in *International Conference on Computer Networks and Information Technology*, 2011, pp. 143–147.
[3] E. R. Djuwitaningrum and M. Apriyani, "Teknik Steganografi Pesan Teks Menggunakan Metode Least Significant Bit dan Algoritma Linear Congruential Generator," *JUITA J. Inform.*, vol. 4, no. 2, pp. 79–85, 2017.
[4] R. Munir, *Kriptografi*, Edisi Kedua. Bandung: Informatika, 2019.
[5] R. Arifin and L. T. Oktoviana, "Implementasi Kriptografi dan Steganografi menggunakan Algoritma RSA dan metode LSB," *Univ. Malang*, 2013.
[6] B. Isnanto and A. A. Alkodri, "Kriptografi DES Dan Steganografi Pada Dokumen Dan Citra Digital Menggunakan Metode LSB," *J. TI Atma Luhur*, vol. 1, no. 1, pp. 38–45, 2014.
[7] G. W. Bhaudhayana and I. M. Widiartha, "Implementasi algoritma kriptografi aes 256 dan metode steganografi lsb pada gambar bitmap," *J Ilmu Komput*, vol. 8, no. 2, pp. 15–25, 2015.
[8] I. M. Arrijal, R. Efendi, and B. Susilo, "Penerapan Algoritma Kriptografi Kunci Simetris Dengan Modifikasi Vigenere Cipher Dalam Aplikasi Kriptografi Teks," *Pseudocode*, vol. 3, no. 1, pp. 69–82, 2016.
[9] A. Rohmanu, "Implementasi kriptografi dan steganografi dengan metode algoritma DES dan metode End Of File," *J. Inform. SIMANTIK*, vol. 2, no. 1, pp. 1–11, 2017.
[10] D. Darwis, W. Wamiliana, and A. Junaidi, "Proses Pengamanan Data Menggunakan Kombinasi Metode Kriptografi Data Encryption Standard dan Steganografi End Of File," in *Prosiding Seminar Nasional METODE KUANTITATIF 2017*, 2017, vol. 1, no. 1, pp. 228–240.
[11] I. Utomo, W. S. Sari, and C. A. Sari, "Kombinasi Steganografi-Kriptografi Citra Menggunakan LSB Dan DES," *SNATIF*, vol. 5, no. 1, 2018.
[12] P. H. Rantelinggi and E. Saputra, "Algoritma Kriptografi Triple Des dan Steganografi LSB sebagai Metode Gabungan dalam Keamanan Data," *J. Teknol. Inf. Dan Ilmu Komput.*, vol. 7, no. 4, pp. 661–666, 2019.
[13] I. Febriana, "PENERAPAN TEKNIK KRIPTOGRAFI PADA KEAMANAN SMS ANDROID," *JoEICT J. Educ. ICT*, vol. 1, no. 1, Art. no. 1, Mar. 2017, doi: 10.29100/.v1i1.103.
[14] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "Digital image steganography: Survey and analysis of current methods," *Signal Process.*, vol. 90, no. 3, pp. 727–752, 2010.

ISSN: 2337-7631 (printed)
ISSN: 2654-4091 (online)