Copyright © The Author(s)
This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License

JUSIFO JURNAL SISTEM INFORMASI

p-ISSN: 2460-092X, e-ISSN: 2623-1662
Vol. 6, No. 2, Desember 2020
Hal. 111 - 120

Peningkatan Keamanan Pengiriman Pesan Teks: Kombinasi *Advanced Encryption Standard (AES) 128* dan *Least Significant Bit (LSB)*

Yulia Fatma[1*], Afdhil Hafid[2], Heru Oktavian Dani[1]
yuliafatma@umri.ac.id[*]

[1]Universitas Muhammadiyah Riau - Indonesia
[2]Universitas Islam Negeri Imam Bonjol Padang - Indonesia

Diterima: 16 Okt 2020 | Direvisi: 08 Nov - 24 Nov 2020
Disetujui: 19 Des 2020 | Dipublikasi: 28 Des 2020
Program Studi Sistem Informasi, Fakultas Sains dan Teknologi,
Universitas Islam Negeri Raden Fatah Palembang, Indonesia

ABSTRACT

Confidentiality in transfer the messages is an important thing to maintain. Increased security in transfer the messages can be improved using cryptography and steganography. This article aims to apply a combination of AES 128 cryptography and LSB steganography in an application, and measures the quality of the stego-image. In this article, the AES 128 algorithm is used for the encryption process of secret messages and the LSB algorithm for the process of inserting secret messages in the cover image. PSNR is used to measure the quality of the stego-image, by comparing the cover image and the stego-image. This research produces an application that can be used for the security of sending text messages by combining the AES 128 algorithm in the plaintext encryption and the LSB algorithm in the secret message insertion process. In term of measuring the quality of the stego-image, it was found that the character length of the secret message would have an impact on the MSE and PSNR values. This means, the longer the character of the secret message will affect the quality of the stego-image.
Keywords: *Cryptography, Steganography, Stego-image Quality*

ABSTRAK

Kerahasiaan dalam melakukan pengiriman pesan merupakan hal yang penting untuk dijaga. Peningkatan keamanan dalam pengiriman pesan bisa dilakukan dengan menggunakan kriptografi dan steganografi. Artikel ini bertujuan untuk menerapkan kombinasi kriptografi AES 128 dan steganografi LSB pada suatu aplikasi, dan mengukur kualitas dari stego-image yang dihasilkan. Di dalam artikel ini digunakan algoritma AES 128 untuk proses enkripsi pesan rahasia dan algoritma LSB untuk proses penyisipan pesan rahasia pada gambar penampung. Untuk mengukur kualitas stego-image digunakan PSNR, dengan membandingkan gambar asli dan stego-image. Dari penelitian ini menghasilkan aplikasi yang dapat digunakan untuk keamanan pengiriman pesan teks dengan mengkombinasikan algoritma AES 128 pada enkripsi pesan dan algoritma LSB pada proses penyisipan pesan rahasia (ciphertext), dan pada pengukuran kualitas stego-image didapatkan bahwa panjang karakter pesan rahasia akan berdampak pada nilai MSE dan PSNR. Hal ini brarti, semakin panjang karakter pesan rahasia akan berpengaruh pada kualitas stego-image yang dihasilkan.
Kata Kunci: *Kriptografi, Steganografi, Kualitas Stego-image*

PENDAHULUAN

Berkembangnya berbagai media komunikasi yang didukung dengan semakin meluasnya jaringan *internet* membawa masyarakat semakin dimudahkan (Putra, 2019). Keamanan informasi melalui jaringan *internet* menjadi hal yang perlu diperhatikan. Pengamanan yang kurang baik dapat mengakibatkan informasi tersebut dapat diakses

oleh pihak yang tidak berwenang, akibatnya pemilik informasi dapat mengalami kerugian (Gunawan & Fenando, 2018). Di sisi lain, kerahasiaan dalam melakukan pengiriman pesan juga merupakan hal yang penting untuk dijaga. Peningkatan keamanan dalam pengiriman pesan bisa dilakukan dengan menggunakan kriptografi dan steganografi (Fatma et al., 2018). Kriptografi dalam hal ini dapat digunakan untuk mengubah suatu pesan asli menjadi tidak bisa dimengerti. Sedangkan steganografi dapat digunakan untuk menyisipkan pesan rahasia tersebut ke dalam suatu media penampung (Bhaudhayana & Widiartha, 2015).

Pada penelitian (Anwar et al., 2018; Patil et al., 2016), disebutkan bahwa *Advanced Encryption Standard (AES)* merupakan algoritma kriptografi yang paling cocok digunakan pada suatu aplikasi pengamanan pengiriman pesan rahasia, dan memiliki kinerja waktu pemrosesan yang lebih cepat. Pada penelitian (Laurentinus et al., 2020), dijelaskan bahwa *AES* memiliki kinerja waktu enkripsi dan dekripsi rata-rata setiap karakter pada *AES* lebih cepat jika dibandingkan dengan algoritma lain, seperti *Rivest Shamir Adleman (RSA)*. Peningkatan keamanan dalam pengiriman pesan rahasia juga dapat dilakukan dengan menggabungkan antara kriptografi dengan steganografi. Media penampung yang digunakan pada steganografi dapat bermacam-macam. Salah satu media penampungan yang dapat dengan mudah ditemukan yaitu gambar. Pemanfaatan gambar sebagai media penampung pesan rahasia pada steganografi sudah pernah dilakukan pada penelitian (Abdel Wahab et al., 2019; Darbani et al., 2019; Fatma & Wardoyo, 2017). Gambar memiliki nilai yang kaya akan informasi, sehingga penyisipan pesan rahasia pada gambar tidak akan terlihat secara kasat mata oleh manusia. Pada penelitian (Antonio, 2013), dijelaskan bahwa penggunaan steganografi *Least Significant Bit (LSB)* akan menghasilkan *stego-image* yang tidak jauh berbeda dengan gambar aslinya sebelum dilakukan penyisipan. Hal ini juga didukung dengan penelitian (Susanto & Mulyono, 2020) yang memperlihatkan hasil histogram untuk gambar asli dan *stego-image* dengan menggunakan *LSB* tidak berbeda jauh.

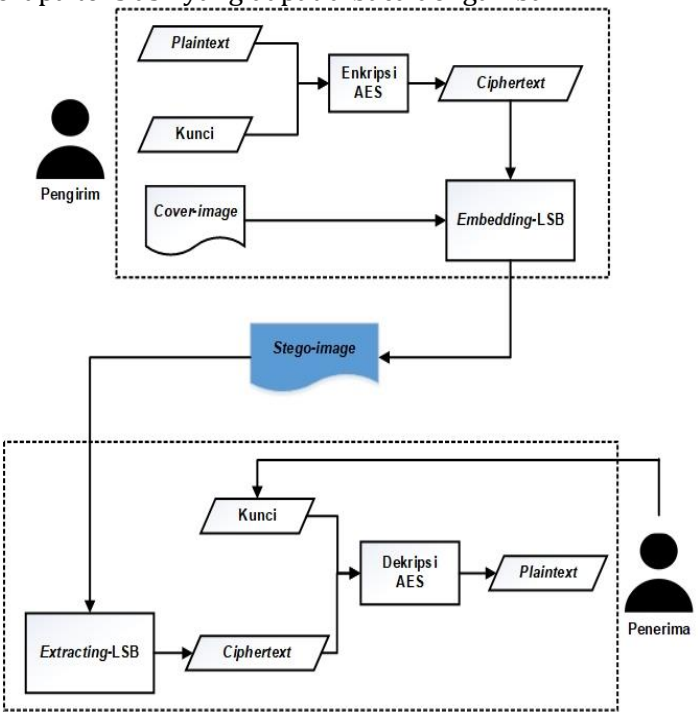
Artikel ini bertujuan untuk menerapkan kombinasi kriptografi *AES 128* dan steganografi *LSB* pada suatu aplikasi, dan mengukur kualitas dari *stego-image* yang dihasilkan. Untuk mengukur kualitas *stego-image* digunakan *Peak Signal to Noise Ratio (PSNR)* (Ansari et al., 2020; Widyawati, 2019). Dengan menggunakan *PSNR*, gambar asli akan dibandingkan dengan *stego-image* (Kim et al., 2019).

METODOLOGI PENELITIAN

Pada penelitian ini digunakan pendekatan *Research and Development (R&D)*. Langkah awal sebelum membangun aplikasi dilakukan studi literatur terkait dengan penggunaan algoritma *AES 128* untuk proses kriptografi pesan. Selanjutnya, referensi terkait algoritma *LSB* untuk proses steganografi juga dikumpulkan dan dipelajari. Hal ini dilakukan untuk menghasilkan aplikasi yang sesuai dengan kebutuhan. Secara sederhana skema perancangan aplikasi dapat dilihat pada Gambar 1. Di dalam skema aplikasi, antara pengirim dan penerima pesan harus menggunakan aplikasi yang sama. Pihak pengirim harus melakukan enkripsi terlebih dahulu sebelum mengirimkan pesan rahasia dan memasukkan suatu kunci yang berupa angka rahasia. Selanjutnya aplikasi akan memproses pesan asli *(plaintext)* menjadi pesan rahasia *(ciphertext)*. Pengirim juga akan diminta untuk memilih suatu gambar yang akan digunakan untuk menyisipkan pesan rahasia tersebut. Hasil keluaran dari aplikasi untuk fitur enkripsi berupa *stego-image* yang secara kasat mata tidak memiliki perbedaan dengan gambar asli sebelum dilakukan enkripsi. *Stego-image* dapat dikirimkan ke penerima menggunakan aplikasi lain, seperti: *Microsoft Outlook*, *Whatsapp*, *Google Mail*.

Kemudian, untuk melihat pesan rahasia, maka penerima harus menggunakan fitur dekripsi pada aplikasi. Langkah pertama penerima harus memasukkan *stego-image*.

Ketika *Extracting LSB* selesai diproses oleh aplikasi, kemudian penerima juga harus memasukkan kunci yang telah disepakati dengan pengirim. Keluaran dari aplikasi pada sisi penerima berupa teks asli yang dapat dibaca dengan baik.



Gambar 1. Skema Rancangan Aplikasi

Proses Enkripsi dan *Embedding* Pesan Rahasia

Pada penelitian ini digunakan *activity diagram* untuk menggambarkan proses enkripsi dan *embedding* pesan rahasia. Proses enkripsi pada aplikasi digunakan *AES 128* untuk mengubah *plaintext* menjadi *ciphertext*. Selanjutnya pengirim memilih *cover-image* (gambar asli sebelum disisipkan pesan rahasia), proses penyisipan *(embedding) ciphertext* dilakukan dengan menggunakan *LSB*. Hasil keluaran aplikasi di sisi pengirim berupa *stego-image*. Dapat dilihat pada Gambar 2.

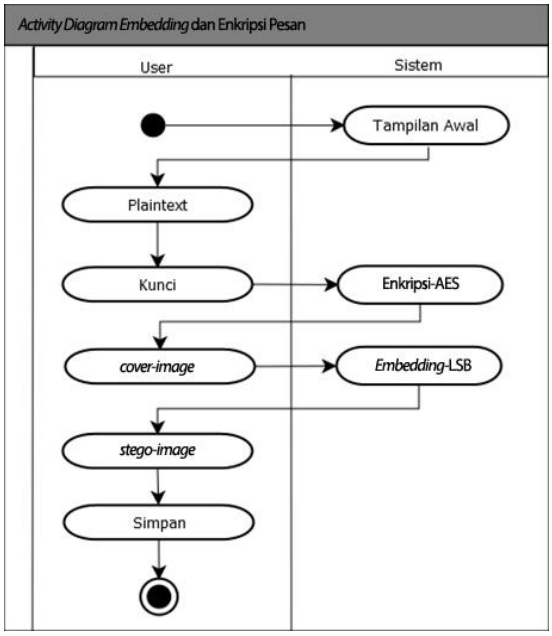
Proses Dekripsi dan *Extracting* Pesan Rahasia

Pada sisi penerima, *stego-image* yang telah diterima dari pengirim kemudian dilakukan proses ekstraksi *(extracting)* menggunakan algoritma *LSB* sehingga mendapatkan *ciphertext*. Penerima pesan melakukan proses dekripsi pada *ciphertext* menggunakan algoritma *AES* dan kunci rahasia yang sudah disepakati sebelumnya dengan pengirim. Hasil keluaran aplikasi di sisi penerima berupa *plaintext* yang dapat dibaca oleh penerima. Dapat dilihat pada Gambar 3.

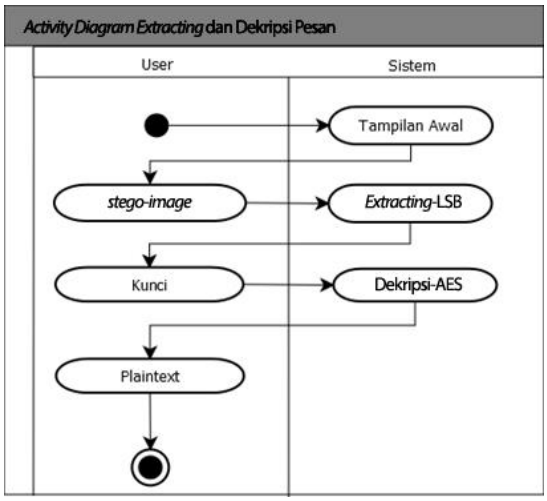
HASIL DAN PEMBAHASAN

Implementasi Tampilan Aplikasi

Tampilan halaman depan dari aplikasi pada penelitian ini dapat dilihat pada Gambar 4. Pada halaman depan ini, pengguna dapat memilih menu yang tersedia, yaitu proses enkripsi atau proses dekripsi. Menu enkripsi dapat digunakan oleh pengirim untuk menghasilkan *stego-image*, sedangkan menu dekripsi dapat digunakan oleh penerima untuk membuka dan membaca pesan rahasia dari pengirim.



Gambar 2. *Activity Diagram* Proses Enkripsi dan *Embedding* Pesan

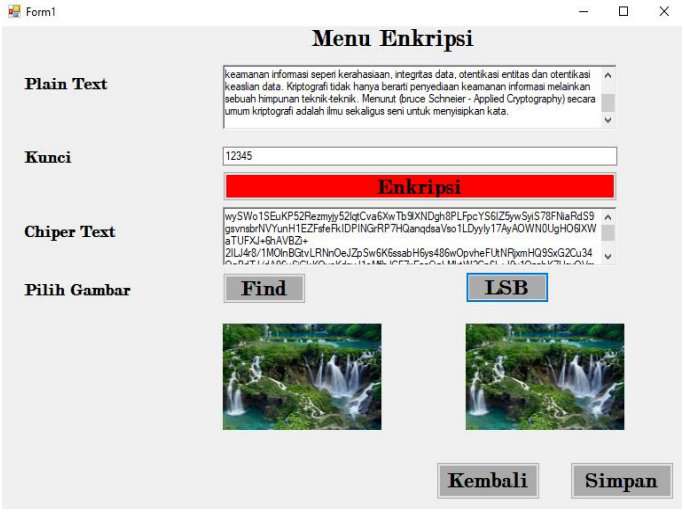


Gambar 3. *Activity Diagram* Proses Dekripsi dan *Extracting* Pesan

Gambar 4. Tampilan Halaman Depan

Pada saat pengirim memilih menu enkripsi, akan tersedia beberapa kolom isian seperti *plaintext*, kunci, *ciphertext*, dan pilih gambar. Kolom *ciphertext* akan ditampilkan

untuk melihat bahwa proses enkripsi kriptografi berhasil dilakukan. Untuk melakukan proses steganografi, diperlukan gambar berwarna yang digunakan sebagai gambar penampung. Kemudian *ciphertext* akan diubah menjadi bentuk *bit* dan digabungkan ke dalam nilai *pixel* gambar dengan menggunakan algoritma *LSB*. Hasil keluaran dari proses enkripsi berupa *stego-image* yang berisi pesan rahasia di dalamnya. Tampilan menu enkripsi dapat dilihat pada Gambar 5.



Gambar 5. Tampilan Menu Enkripsi dan *Embedding* Pesan Rahasia

Proses selanjutnya di sisi penerima, yaitu proses ekstraksi *stego-image*. Dari proses ekstraksi *stego-image* ini akan menghasilkan *ciphertext*. Selanjutnya, untuk dapat membaca pesan rahasia tersebut, diperlukan proses dekripsi. Pada proses dekripsi ini, penerima diminta untuk memasukkan kunci rahasia yang telah disepakati. Hasil keluaran berupa *plaintext* dapat dilihat pada kolom hasil pada aplikasi, dapat dilihat pada Gambar 6.

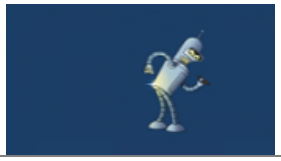
Gambar 6. Tampilan Menu Dekripsi dan *Extracting* Pesan

Pengujian Proses Kriptografi *AES 128* dan Steganografi *LSB*

Pengujian yang dilakukan meliputi pengujian enkripsi, pengujian *embedding* pesan, pengujian *extracting* pesan dan pengujian dekripsi. Pengujian kualitas *stego-image* yang

dihasilkan aplikasi, digunakan *PSNR*. Gambar penampung *(cover image)* yang digunakan dapat dilihat pada Tabel 1.

Tabel 1. Gambar Penampung yang Digunakan

No.	Gambar	Nama Gambar	Dimensi	*Size*
1		Air terjun.png	200 x 200	86.4 Kb
2		Robot.png	200 x 112	24.0 Kb

Sumber: www.google.com

1) Pengujian Enkripsi

Pada proses enkripsi dapat dilihat penggunaan *RAM* dan *CPU* semakin meningkat secara teratur seiring panjangnya pesan yang diproses, kecuali pada waktu proses yang terlihat tetap stabil. Hasil pengamatan pada proses enkripsi dapat dilihat pada Tabel 2.

Tabel 2. Hasil Pengujian Proses Enkripsi

Gambar	Panjang Pesan	*RAM*	*CPU*	Waktu Proses	Ket.
1	5000 Karakter	17.4 Mb	0.3%	0.1 Detik	Berhasil
1	11.241 Karakter	18.8 Mb	0.4%	0.1 Detik	Berhasil
1	11.293 Karakter	18.20 Mb	0.5%	0.1 Detik	Berhasil
2	5.000 Karakter	17.4 Mb	0.3%	0.1 Detik	Berhasil
2	6.282 Karakter	17.9 Mb	0.4%	0.1 Detik	Berhasil
2	6.288 Karakter	17.9 Mb	0.4%	0.1 Detik	Berhasil

2) Pengujian Penyisipan *(embedding)* Pesan

Ciphertext yang dihasilkan dari proses enkripsi diubah menjadi *bit* untuk dimasukkan ke dalam *pixel* gambar dengan menggunakan metode *LSB*. Hasil pengamatan pada proses penyisipan pesan dapat dilihat pada Tabel 3.

Tabel 3. Hasil Pengujian Proses Penyisipan Pesan Rahasia

Gambar	Panjang Pesan	*RAM*	*CPU*	Waktu Proses	*Size*	Ket.
1	5000 Karakter	18.5 Mb	2.1%	0.7 Detik	109 Kb	Berhasil
1	11.241 Karakter	24.9 Mb	2.5%	1.1 Detik	109 Kb	Berhasil
1	11.293 Karakter	24.1 Mb	2.5%	1.4 Detik	109 Kb	Berhasil
2	5.000 Karakter	18.7 Mb	2.1%	0.6 Detik	24.4 Kb	Berhasil
2	6.282 Karakter	23.1 Mb	3.1%	0.7 Detik	27 Kb	Berhasil
2	6.288 Karakter	23.1 Mb	3.1%	0.7 Detik	27 Kb	Berhasil

Pengujian yang dilakukan pada proses penyisipan *chiphertext* ke dalam gambar memperlihatkan rata-rata penggunaan *RAM*, *CPU* dan waktu proses yang meningkat. Dari pengamatan ini dapat dilihat bahwa panjang *ciphertext* yang disisipkan ke dalam gambar mempengaruhi *RAM*, *CPU* dan waktu proses penyisipan pesan rahasia pada gambar.

3) Pengujian Ekstraksi *(extracting)* Pesan Rahasia

Dari hasil pengamatan yang ditampilkan pada Tabel 4 dapat dilihat bahwa panjang *ciphertext* yang terdapat pada gambar mempengaruhi waktu proses dalam melakukan ekstraksi. Pengggunaan *RAM* dan *CPU* menunjukkan pola yang meningkat pada setiap pengujian.

Tabel 4. Hasil Pengujian Ekstraksi Pesan Rahasia

Gambar	Panjang Pesan	*RAM*	*CPU*	Waktu Proses	Ket.
1	5000 Karakter	24.0 Mb	2.1%	10.7 Detik	Berhasil
1	11.241 Karakter	27.9 Mb	2.6%	29.3 Detik	Berhasil
1	11.293 Karakter	28.2 Mb	2.7%	30.1 Detik	Berhasil
2	5.000 Karakter	24.0 Mb	2.1%	10.7 Detik	Berhasil
2	6.282 Karakter	26.7 Mb	2.5%	11.3 Detik	Berhasil
2	6.288 Karakter	26.7 Mb	2.5%	11.4 Detik	Berhasil

4) Pengujian Dekripsi

Proses dekripsi merupakan proses mengubah *ciphertext* yang telah diekstrak dari gambar menjadi *plaintext*. Hasil pengamatan proses dekripsi dapat dilihat pada Tabel 5.

Tabel 5. Hasil Pengujian Dekripsi

Gambar	Plaintext	Waktu Proses	Akurasi	Ket.
1	5000 Karakter	0.1 Detik	100%	Berhasil
1	11.241 Karakter	0.1 Detik	100%	Berhasil
1	11.293 Karakter	0.1 Detik	0%	Gagal
2	5.000 Karakter	0.1 Detik	100%	Berhasil
2	6.282 Karakter	0.1 Detik	100%	Berhasil
2	6.288 Karakter	0.1 Detik	0%	Gagal

Dari hasil pengujian yang dilakukan pada proses dekripsi memiliki tingkat akurasi sebesar 100%, kecuali pada baris 3 dan baris 6. Pada gambar dengan nama air terjun.png dengan dimensi 200 x 200 *pixels* memiliki ambang batas tampung maksimal 11.241 karakter *(plaintext)*, sedangkan untuk gambar dengan nama robot.png dengan dimensi 200 x 112 *pixels* memiliki ambang batas tampung maksimal 6.282 karakter *(plaintext)*. Kegagalan proses dekripsi tersebut disebabkan oleh dimensi gambar yang tidak cukup untuk menampung seluruh *plaintext* yang telah disisipkan. Pada proses dekripsi membutuhkan ruang (dimensi gambar) yang lebih besar jika dibandingkan dengan proses enkripsi yang tidak pernah mengalami kegagalan.

Pengukuran Kualitas Gambar

Pengukuran kualitas gambar dilakukan dengan menggunakan rumus *PSNR*. Gambar asli akan dibandingkan dengan *stego-image*. Gambar yang diukur masih menggunakan gambar yang dipakai tahap pengujian yang telah dijelaskan sebelumnya. Untuk mendapatkan nilai *PSNR* maka diperlukan untuk melakukan perhitungan *Mean Square Error (MSE)* terlebih dahulu. Rumus yang digunakan untuk menghitung nilai *MSE* dapat dilihat pada persamaan (1), dan rumus untuk menghitung nilai *PSNR* dapat dilihat pada persamaan (2).

$$MSE = \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} (S_{xy} - C_{xy})^2 \qquad (1)$$

Dengan ketentuan:
x, y : Koordinat suatu titik pada gambar
M, N : Dimensi dari gambar
S : Gambar tersisipi *(stego-image)*
C : Gambar asli *(cover image)*

$$PSNR = 10.\log_{10}\left(\frac{C_{\max}^2}{MSE}\right) \qquad (2)$$

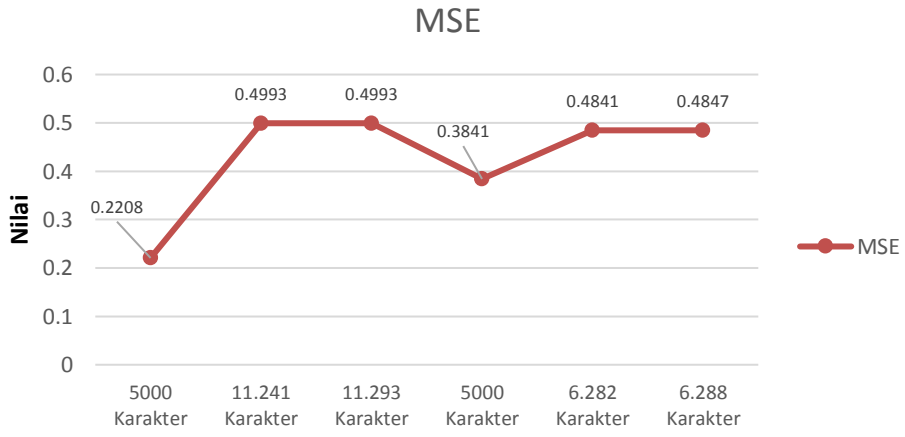
Dengan ketentuan:
C_{max} : Nilai *pixel* terbesar pada keseluruhan gambar
C : Gambar asli *(cover image)*

Selanjutnya, pada Tabel 6 ditampilkan hasil perhitungan *MSE* dan *PSNR*. Dari hasil ini, dapat diketahui bahwa panjang pesan rahasia berpengaruh terhadap nilai *MSE* dan *PSNR*. Pada baris 3 dan 6 untuk nilai *MSE* dan *PSNR* yang diperoleh tidak begitu jauh berbeda, hal ini disebabkan panjang karakter pesan untuk gambar baris ke 2 dan 3 tidak jauh berbeda, demikian juga untuk baris ke 5 dan 6 tidak terlalu jauh berbeda panjang karakter pesan yang dimasukkan.

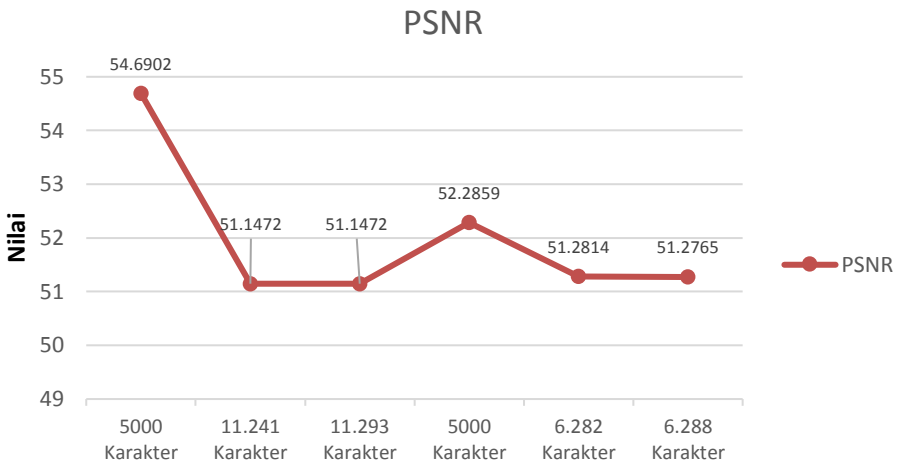
Pada Gambar 7 diperlihatkan hasil perhitungan nilai *MSE* dalam bentuk grafik. Gambar yang disisipkan karakter pesan yang panjang maka akan menghasilkan nilai *MSE* yang tinggi pula. Sedangkan pada Gambar 8 memperlihatkan gambar yang disisipkan karakter pesan yang panjang maka akan menghasilkan nilai *PSNR* yang semakin rendah. Nilai *PSNR* yang rendah menunjukkan kualitas *stego-image* yang rendah juga.

Tabel 6. Hasil Perhitungan *MSE* dan *PSNR*

Gambar	Panjang Pesan	*MSE*	*PSNR*
1	5000 Karakter	0.2208	54.6902
1	11.241 Karakter	0.4993	51.1472
1	11.293 Karakter	0.4993	51.1472
2	5000 Karakter	0.3841	52.2859
2	6.282 Karakter	0.4841	51.2814
2	6.288 Karakter	0.4847	51.2765



Gambar 7. Grafik Nilai *MSE* Terhadap Panjang Pesan



Gambar 8. Grafik Nilai *PSNR* Terhadap Panjang Pesan

KESIMPULAN

Penelitian ini telah menghasilkan aplikasi yang dapat digunakan untuk keamanan pengiriman pesan teks dengan mengkombinasikan algoritma *AES 128* pada proses enkripsi dan algoritma *LSB* pada proses penyisipan pesan rahasia *(ciphertext)*. Keberhasilan proses dekripsi pesan rahasia sangat dipengaruhi oleh dimensi gambar dan panjang karakter. Semakin tinggi ukuran dimensi gambar, maka semakin panjang karakter pesan yang dapat ditampung. Setiap gambar memiliki ambang batas tampung dari panjang karakter pesan rahasia yang disisipkan terhadap dimensi gambar.

Pengukuran kualitas *stego-image* telah dilakukan dan didapatkan kesimpulan bahwa panjang karakter pesan rahasia akan berdampak pada nilai *MSE* dan *PSNR*. Hal ini brarti, semakin panjang karakter pesan akan berpengaruh pada kualitas *stego-image* yang dihasilkan.

DAFTAR RUJUKAN

Abdel Wahab, O. F., Hussein, A. I., Hamed, H. F. A., Kelash, H. M., Khalaf, A. A. M., & Ali, H. M. (2019). Hiding Data in Images Using Steganography Techniques with Compression Algorithms. *Telkomnika (Telecommunication Computing Electronics and Control)*, *17*(3), 1168–1175. https://doi.org/10.12928/TELKOMNIKA.V17I3.12230

Ansari, A. S., Mohammadi, M. S., & Parvez, M. T. (2020). a Multiple-format Steganography Algorithm for Color Images. *IEEE Access*, *8*, 83926–83939. https://doi.org/10.1109/ACCESS.2020.2991130

Antonio, H. (2013). Studi Perbandingan Enkripsi Steganografi dengan Menggunakan Metode Least Significant Bit dan End of File. *JUSTIN (Jurnal Sistem dan Teknologi Informasi)*, *1*(2), 132–137.

Anwar, N., Munawwar, M., Abduh, M., & Santosa, N. B. (2018). Komparatif Performance Model Keamanan Menggunakan Metode Algoritma AES 256 bit dan RSA. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, *2*(3), 783–791. https://doi.org/10.29207/resti.v2i3.606

Bhaudhayana, G. W., & Widiartha, I. M. (2015). Implementasi Algoritma Kriptografi AES 256 dan Metode Steganografi LSB pada Gambar BITMAP. *Jurnal Ilmu Komputer*, *8*(2), 15–25.

Darbani, A., AlyanNezhadi, M. M., & Forghani, M. (2019). A New Steganography Method for Embedding Message in JPEG Images. *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*, 617–621. https://doi.org/10.1109/KBEI.2019.8735054

Fatma, Y., Mukhtar, H., & Taufik, M. (2018). Implementasi Steganografi pada Teks Terenkripsi dengan Algoritma RSA Menggunakan Metode BPCS. *Jurnal Fasilkom*, *7*(2), 260–265. https://doi.org/10.37859/jf.v7i2.783

Fatma, Y., & Wardoyo, R. (2017). *Pembangkitan Kunci Simetri Menggunakan Gambar dan Single Layer Perceptron*. Universitas Gadjah Mada.

Gunawan, C. E., & Fenando, F. (2018). Pengukuran Keamanan Informasi Menggunakan Indeks Keamanan Informasi (KAMI) Studi Kasus di PUSTIPD UIN Raden Fatah Palembang. *JUSIFO (Jurnal Sistem Informasi)*, *4*(2), 121–132. https://doi.org/10.19109/JUSIFO.V4I2.4107

Kim, C., Shin, D., Yang, C.-N., Chen, Y.-C., & Wu, S.-Y. (2019). Data Hiding Using Sequential Hamming + K with M Overlapped Pixels. *KSII Transactions on Internet and Information Systems*, *13*(12), 6159–6174. https://doi.org/10.3837/tiis.2019.12.020

Laurentinus, L., Pradana, H. A., Sylfania, D. Y., & Juniawan, F. P. (2020). Performance comparison of RSA and AES to SMS messages compression using Huffman algorithm. *Jurnal Teknologi dan Sistem Komputer*, *8*(3), 171–177. https://doi.org/10.14710/jtsiskom.2020.13468

Patil, P., Narayankar, P., Narayan, D. G., & Meena, S. M. (2016). A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish. *Procedia Computer Science*, *78*, 617–624. https://doi.org/10.1016/j.procs.2016.02.108

Putra, R. A. (2019). Tantangan Media Massa Dalam Menghadapi Era Disrupsi Teknologi Informasi. *JUSIFO (Jurnal Sistem Informasi)*, *5*(1), 1–6. https://doi.org/10.19109/jusifo.v5i1.5003

Susanto, A., & Mulyono, I. U. W. (2020). Kombinasi LSB-RSA untuk Peningkatan Imperceptibility pada Kripto-Stegano Gambar RGB. *Proceeding SENDIU 2020*, *July*, 21–27.

Widyawati, L. (2019). *Implementasi Metode Steganografi SLT-DCT pada Citra untuk Meningkatkan Kualitas Citra Steganografi*.