

Analisis Histogram Steganografi Least Significant Bit Pada Citra Grayscale

Nur Fitrianiingsih Hasan¹, Christin Nandari Dengen², Dony Ariyus³

^{1,2,3}Magister Teknik Informatika, Universitas AMIKOM Yogyakarta, Jalan Ring Road Utara, Condong Catur, Depok, Sleman, Yogyakarta, telp (0274) 884201-207
e-mail: ¹nur.1063@students.amikom.ac.id, ²christin.dengen@students.amikom.ac.id, ³dony.a@amikom.ac.id

Abstrak

Kesadaran pentingnya keamanan data pada era digital saat ini ditandai dengan berkembangnya aplikasi keamanan data. Pada penelitian ini akan dilakukan analisis histogram terhadap suatu aplikasi keamanan data. Tujuan dari analisa ini adalah mengetahui perubahan histogram yang dilakukan penyisipan pesan. Hasilnya penyisipan pesan pada citra menggunakan Least Significant Bit secara visual tidak terlihat perubahan. Analisis histogram menunjukkan perubahan grafik terlihat lebih signifikan pada file gambar .tif, .png dan .jpg sementara .bmp tidak signifikan. Jumlah karakter yang disisipkan mempengaruhi grafik histogram. Stego image yang dihasilkan memenuhi syarat imperceptibility. Dibuktikan dari nilai PSNR yang dihasilkan yaitu PSNR >40db. Perubahan ekstensi image menjadi salah satu faktor hasil nilai PSNR. Dibuktikan dengan ekstensi image .bmp memiliki nilai MSE paling rendah dan PSNR paling tinggi yaitu, MSE=0,66141 dan PSNR=49,9601.

Kata kunci: Histogram, Steganografi, LSB, Grayscale, PSNR

Abstract

Awareness of the importance of data security in the current digital era is marked by the development of data security applications. In this study a histogram analysis of a data security application will be performed. The purpose of this analysis is to determine the histogram changes made by message insertion. The result is the insertion of messages in the image using Least Significant Bit visually does not change. Histogram analysis shows that graph changes are more significant in .tif, .png and .jpg image files while .bmp is not significant. The number of characters inserted affects the histogram graph. The resulting stego image meets the requirements of imperceptibility. Proven from the PSNR value generated is PSNR>40dB. The change in image extension is one of the factors resulting from the PSNR value. Evidenced by the extension .bmp image has the lowest MSE value and the highest PSNR, namely, MSE=0.66141 and PSNR=49.9601.

Keywords: Histogram, Steganography, LSB, Grayscale, PSNR.

1. Pendahuluan

Melindungi data dan informasi perusahaan ataupun organisasi dari upaya *cyber-attack* menjadi sangat esensial bagi suatu organisasi, baik organisasi komersil, perguruan tinggi, lembaga pemerintah maupun individual. Jatuhnya informasi ke tangan pihak lain yang tidak bertanggung jawab dapat menimbulkan kerugian bagi pemilik informasi. Keamanan informasi yang dimaksudkan mencapai tiga sasaran utama yaitu kerahasiaan, integritas dan ketersediaan[1]. Era big data saat ini, informasi dapat dengan mudah di interupsi, disadap, di modifikasi dan di fabrikasi oleh pihak tidak bertanggung jawab[2]. Upaya untuk meminimalisir

cyber-attack tersebut dapat dilakukan dengan memberikan pengamanan terhadap informasi atau data. Selain kriptografi ada satu teknik penyembunyian data yang paling terkenal yaitu steganografi[3][4]. Steganografi merupakan seni dan ilmu yang mempelajari teknik penyembunyian pesan atau informasi pada suatu media gambar atau video dan lainnya. Steganografi berbeda dengan kriptografi, ini karena steganografi menyembunyikan pesan di dalam sebuah objek sedemikian rupa hingga tidak menimbulkan kecurigaan[5]. Konsep steganografi adalah menyisipkan (*embedding*) dan ekstraksi (*extracting*) pada sebuah media citra seperti foto dan video[4][6]. Tujuan dari steganografi adalah agar informasi tidak terdeteksi oleh pihak yang tidak berhak atas informasi tersebut[9].

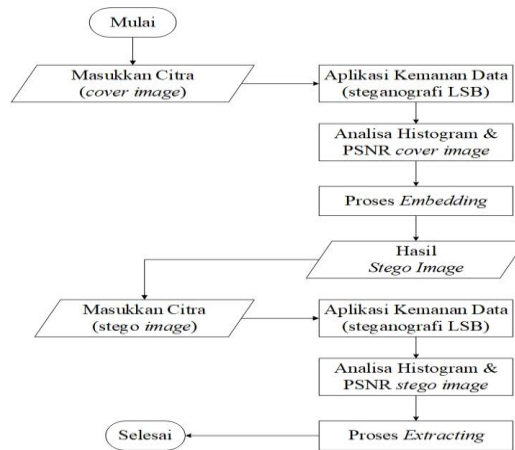
Beberapa penelitian sebelumnya yang berhubungan dengan steganografi sudah pernah dilakukan dengan tujuan, analisa dan pengembangan yang berbeda-beda sesuai pada kebutuhannya. Penelitian [7] menggunakan LSB pada teknik steganografi dengan objek gambar JPEG dan mengkombinasi dengan algoritma kriptografi *vigenere*. Sementara pada penelitian [8] menyembunyikan pesan rahasia di dalam transkrip nilai mahasiswa menggunakan LSB pada objek gambar JPEG. Terakhir penelitian [9] melakukan analisa pada histogram LSB dengan format JPEG. Pada penelitian-penelitian terdahulu yaitu fokus kepada metode pengembangan teknik kriptografi maupun steganografi, pengujian terhadap hasil pengembangan adalah sebatas berhasil atau tidaknya metode yang diterapkan, sementara pada penelitian ini yang dilakukan adalah menguji dan menganalisa hasil dari *stego image* yaitu melalui histogram. Analisa histogram memiliki peran penting dalam gambar steganografi[10].

penelitian ini bertujuan untuk menganalisa visual *cover image* dan *stego image* yang telah di sisipkan pesan rahasia dari pengembangan kriptografi *playfair cipher* dan *least significant bit* (LSB) menggunakan histogram masing-masing dan nilai PSNR pada citra *grayscale* dengan inputan objek gambar .jpg, .png, .bmp dan .tif. Analisa histogram yang akan dilakukan menggunakan pengembangan dari Aplikasi Keamanan Data dengan Modifikasi Kunci Algoritma Kriptografi untuk Pesan Rahasia Pada Gambar dengan Matlab 2018a.

2. Metode Penelitian

Penelitian ini menganalisis *cover image* dan *stego image* yang dihasilkan oleh Aplikasi Keamanan Data dengan Modifikasi Kunci Algoritma Kriptografi untuk Pesan Rahasia Pada Gambar menggunakan algoritma LSB pada Matlab. Analisis yang dilakukan yakni membandingkan gambar sebelum dan setelah proses penyisipan pesan, histogram dari gambar *grayscale* sebelum dan sesudah proses penyisipan pesan kedalam gambar dengan menggunakan aplikasi yang telah dibuat pada Matlab 2018a serta membandingkan nilai PSNR dari gambar hasil proses penyisipan. Setelah dilakukan ketiga pengujian tersebut, maka akan di analisis dan diambil kesimpulan untuk penelitian ini. Pada Gambar 1 adalah alur penelitian yang dilakukan. Secara garis besar penelitian dimulai dengan memasukkan citra atau gambar yang nantinya menjadi *cover image* ke sebuah aplikasi yang telah penulis kembangkan. Penggunaan aplikasi ini untuk melihat histogram citra yang selanjutnya di analisis oleh penulis. Setelah histogram muncul, selanjutnya dilakukan proses penyisipan pesan kedalam citra, jika proses penyisipan berhasil dan selesai maka citra tersebut telah menjadi sebuah *stego image*, kemudian disimpan.

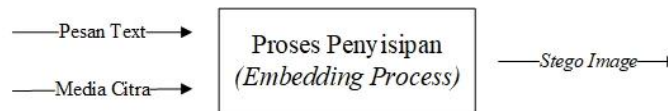
Tahap kedua untuk melihat perbandingan histogram sesudah dan sebelum di lakukan penyisipan maka proses ekstraksi perlu dilakukan, proses ekstraksi adalah proses yang memisahkan pesan rahasida dari sebuah objek (citra). Langkah pertama adalah melakukan input *stego image* yaitu citra yang sebelumnya telah disisipkan pesan rahasia, kemudian aplikasi akan menampilkan histogram, disini penulis melakukan analisa dan membandingkan histogram juga PSNR sebelum dan sesudah citra disisipkan. Analisa histogram yang dilakukan pada *cover image* dan *stego image* selanjutnya di bandingkan dan ditarik sebuah kesimpulan yang akan menjadi sebuah temuan dalam penelitian ini.



Gambar 1. Alur Penelitian

2.1. Proses Penyisipan

Proses dalam steganografi ada 2 yaitu penyisipan (*embedding*) dan ekstraksi (*extracting*) atau memisahkan pesan informasi dari media sisipan. Proses penyisipan dilakukan oleh pengirim pesan dan proses ekstraksi dilakukan oleh penerima pesan[11]. Proses penyisipan atau *embedding process* merupakan proses yang dilakukan untuk menyisipkan pesan ke dalam gambar. Pada penelitian ini yaitu menyisipkan pesan rahasia berupa *text* ke dalam sebuah media citra gambar (*cover image*) untuk menghasilkan sebuah *stego image*, adapun diagram proses penyisipan secara garis besar dapat dilihat pada Gambar 2.



Gambar 2. Proses Penyisipan

Adapun proses dari penyisipan yaitu menggunakan gambar *grayscale* berformat *.bmp* dan pesan yang diinputkan tanpa spasi maupun huruf kapital. *Output* yang dihasilkan dari proses penyisipan yaitu *stego image grayscale* berformat *bmp*. Perancangan algoritma proses penyisipan pesan kedalam *cover image* ditampilkan pada Tabel 1.

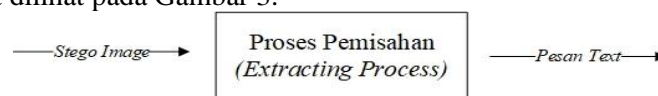
Tabel 1. Proses Penyisipan

Algoritma 1. Proses Penyisipan	
Input	Pesan text dan cover image grayscale (.bmp)
Output	Stego image grayscale (.bmp)
Step 1	Input Kunci
Step 2	Enkripsi Kunci (<i>cipher key</i>)
Step 3	Hasil Cipher key
Step 4	Input Plaintext
Step 5	Enkripsi Plaintext
Step 6	Simpan Ciphertext
Step 7	Input Cover Image
Step 8	Input Ciphertext
Step 9	Create Stego Image
Step 10	Save Stego Images

Pada tabel 1 menjelaskan proses penyisipan pesan pada gambar, dimana langkah pertama yang dilakukan yakni input kunci dan melakukan enkripsi sehingga menghasilkan *cipherkey*. Kemudian dilakukan input plaintext dan dilakukan enkripsi sehingga menghasilkan *ciphertext* lalu disimpan dengan format txt. Lalu masuk dalam proses penyisipan pesan, dimana langkah pertama dilakukan *input cover image* dan *secret message* (*ciphertext*), kemudian dilakukan “*create stego image*” yang berfungsi untuk menyatukan gambar dan pesan yang telah di enkripsi. Lalu akan menghasilkan *stego image* dan disimpan dengan *grayscale* dan format (.bmp). Tahap penyisipan ini dimulai dengan mengubah *pixel*, kemudian bilangan *biner* pada setiap *pixel* di ambil bit rendah akhirnya yaitu 1 *bit* terakhir[12][13].

2.2. Proses Ekstraksi

Perancangan proses ekstraksi merupakan proses yang berfungsi untuk mengekstraksi gambar (*stego image*) dalam format bmp terhadap pesan teks yang telah disembunyikan. Hasil dari proses ini yakni berupa pesan teks hasil penyisipan. Adapun proses atau konsep pemisaahan secara garis besar dapat dilihat pada Gambar 3.



Gambar 3. Proses Ekstraksi

Perancangan algoritma proses ekstraksi pesan teks dari *stego image* ditampilkan pada tabel 2.

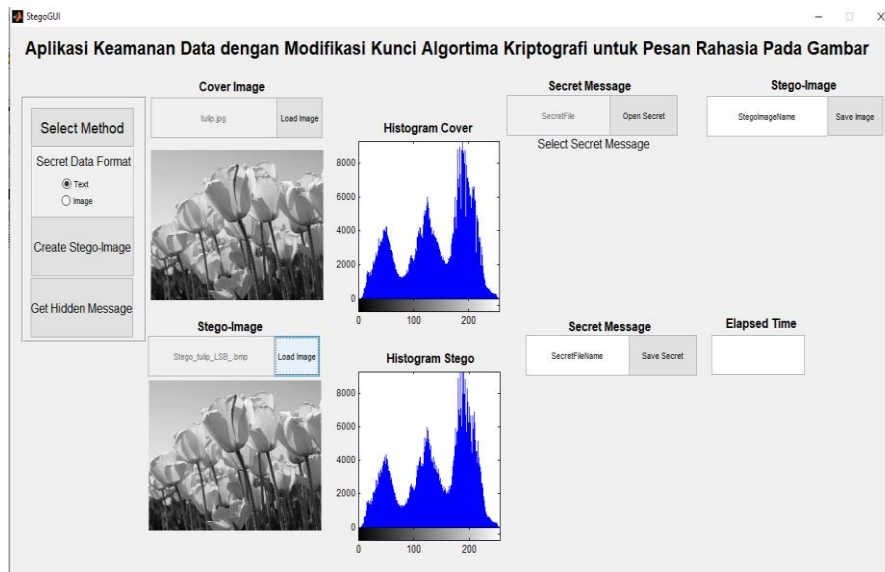
Tabel 2. Proses Ekstraksi

Algoritma 2. Proses Ekstraksi	
Input	Stego Image Grayscale (.bmp)
Output	Cover Image Grayscale (.bmp) dan pesan teks hasil penyisipan
Step 1	Input Cover Image
Step 2	Proses ekstraksi gambar dengan pesan teks
Step 3	Simpan pesan yang telah dipisahkan (<i>chipertext</i>)
Step 4	Input <i>ciphertext</i> hasil ekstraksi
Step 5	Input kunci aviation english
Step 6	Deskripsi pesan (<i>ciphertext</i> menjadi plaintext)
Step 7	Plaintext

Pada tabel 2 di jelaskan mengenai proses ekstraksi, dimana langkah pertama yang dilakukan yakni melakukan *input stego image* (gambar hasil penyisipan pesan). Selanjutnya, dilakukan “*get hidden message*” (proses ekstraksi) pemisahan gambar dengan pesan yang telah disisipkan kemudian pesan tersebut disimpan dengan format txt. Setelah itu, dilakukan input *ciphertext* hasil ekstraksi dan kunci aviation English. Kemudian dilakukan proses deskripsi sehingga menghasilkan plaintext seperti sebelumnya.

2.3. Hardware dan Software

Pada penelitian ini hardware atau perangkat keras yang digunakan sebagai pendukung yaitu Notebook ASUS dengan harddisk 1TB, processor Intel® Core(TM) i3-7020U CPU @2.30GHz, RAM 4.00GB. Adapun software yang digunakan yaitu Matlab 2018a dan Aplikasi Keamanan Data dengan Modifikasi Kunci Algoritma Kriptografi untuk Pesan Rahasia Pada Gambar.



Gambar 4. Tampilan UI Aplikasi Keamanan Data Kriptografi dan Steganografi

Tampilan aplikasi yang digunakan pada penelitian ini adalah aplikasi keamanan data yang dirancang dan dikembangkan oleh penulis. Pada Gambar 4 dapat dilihat tampilan UI pada aplikasi tersebut. Histogram citra yang di *inputkan* oleh *user* akan terbaca, baik citra *cover* maupun *stego*, dari histogram itulah yang akan dijadikan bahan analisis pada penelitian ini yaitu dengan menganalisis visual histogram, nilai MSE dan nilai PSNR.








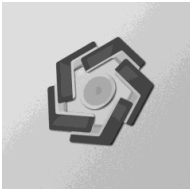


3. Hasil dan Pembahasan

Pada penelitian ini dilakukan pengujian terhadap 5 citra atau gambar sebelum (*cover image*) dan setelah dilakukan penyisipan pesan (*stego image*) menggunakan algoritma LSB. Gambar yang digunakan didapat dari beberapa sumber yang kemudian disesuaikan dengan skenario uji yaitu gambar diubah menjadi berdimensi 512x512 *pixel* dan format gambar dibuat lebih bervariasi bertujuan untuk menguji hasil perubahan citra dalam berbagai format. Skenario pengujian pertama mengacu pada penelitian [10]. Tabel 3 menunjukkan perbandingan secara visual citra sebelum dan sesudah proses *embedd* dengan ukuran kapasitas maksimal. Perbandingan *cover image* dan *stego image* dapat dilihat pada Tabel 3.

Pada Tabel 3 dapat dilihat bahwa citra sebelum dan sesudah dilakukan penyisipan pesan yang sama yaitu dengan kapasitas penyisipan yang sama menggunakan LSB berhasil disisipkan. Hasil uji visual yang dilakukan pada penelitian ini serupa dan mengacu pada penelitian [10] dimana seharusnya *stego image* secara visual sama dengan *cover image*.

Algoritma Kriptografi untuk Pesan Rahasia Pada Gambar dapat menerima jenis file gambar dengan format bervariasi yaitu .TIF, .BMP, .JPG/JPEG, .PNG dengan dimensi 512x512 *pixel* pada penelitian ini dan akan memberikan *output* berupa gambar stego dengan format .BMP. Perubahan yang terjadi dari 5 skenario uji yaitu pada ukuran setiap *file*, dimana ukuran *file* gambar mengalami peningkatan dan penurunan ukuran dari gambar asli yaitu 257 kb, namun secara visual (kasat mata) tidak ada perubahan warna ataupun *noise* yang terlihat. Pada gambar lena.tif 256 kb setelah disisipkan pesan menjadi lena.bmp 257 kb, gambar cameraman.tif 256 kb menjadi cameraman.bmp 257 kb, gambar apel.bmp 512 kb menjadi apel.bmp 257 kb, amikom.png 355 kb menjadi amikom.bmp 257 kb dan tulip.jpg 228 kb menjadi tulip.bmp 257 kb.


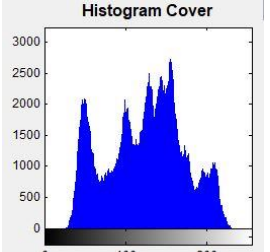
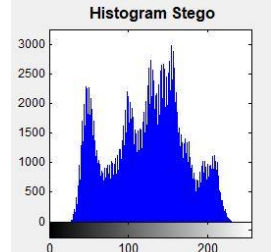

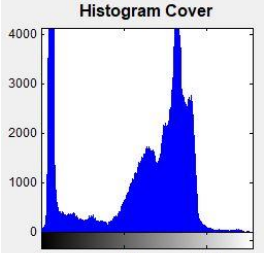
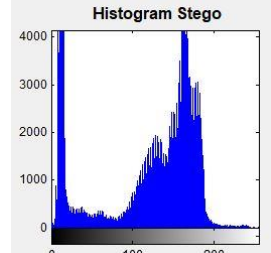

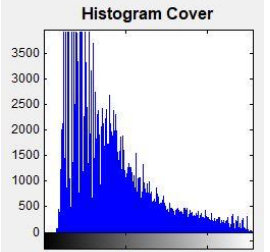
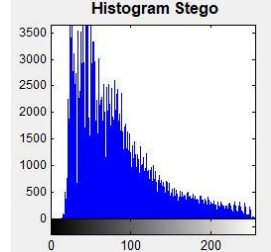
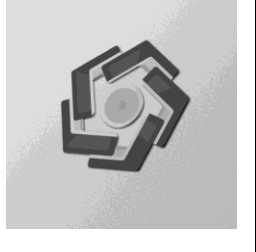
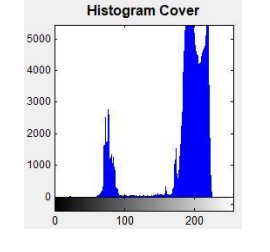
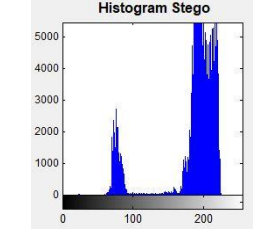
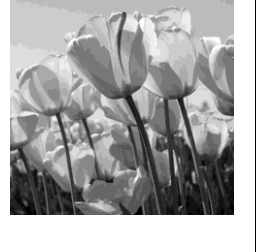
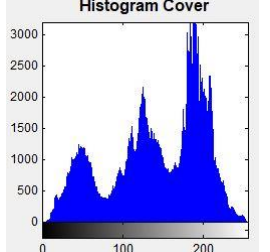
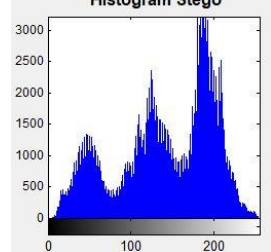
Tabel 3. Perbandingan Visual Citra

No.	Cover Image	Stego Image	Sumber Cover Image
1	 <p>lena.tif 512x512 pixel, 256 kb</p>	 <p>lena.bmp 512x512 pixel, 257 kb</p>	<p>https://4.bp.blogspot.com/-4MoXLSKiTW0/UFdHhDPvFoI/AAAAAAAAAFs/cISSPL2wk5k/s1600/citra+lena+a.PNG</p>
2	 <p>cameraman.tif 512x512 pixel, 256 kb</p>	 <p>cameraman.bmp 512x512 pixel, 257 kb</p>	<p>https://camo.githubusercontent.com/9b80f236dc73c7e12629b06711cc730449023e78/68747470733a2f2f7261772e6769746875622e636f6d2f616e74696d617474657231352f636166572616d616e2f6d61737465722f63616d6572616d616e2e706e67</p>
3	 <p>apel.bmp 512x512 pixel, 512 kb</p>	 <p>apel.bmp 512x512 pixel, 257 kb</p>	<p>https://pbs.twimg.com/media/C4tk4K_UkAQHbta.jpg</p>
4	 <p>amikom.png 512x512 pixel, 355 kb</p>	 <p>amikom.bmp 512x512 pixel, 257 kb</p>	<p>https://1.bp.blogspot.com/-HGJ2mFxmGQ/WBXXmOod2dl/AAAAAAAAAiQ/Z6chtDRgNDgRZkHeQQXMmKqSqkeR3DNyAClB/s1600/logo%2Bamikom%2B5.png</p>
5	 <p>tulip.jpg 512x512 pixel 228 kb</p>	 <p>tulip.bmp 512x512 pixel 257 kb</p>	<p>https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcS9UH5Un8smyFleJlKc9Qua7eQtdjd2Wqg1HdS470Lm-nKj5GIn&s</p>

3.1. Analisis Histogram

Security check pada penelitian ini juga menggunakan *pixel histogram different* (PHD). PHD merupakan teknik evaluasi parameter gambar, tekniknya yaitu mengambil perbedaan antara *cover image* dengan *stego image*[14]. Hasil analisa histogram dapat dilihat pada Tabel 4.

Tabel 4. Perbandingan Histogram

No	Image	Histogram Cover Image	Histogram Stego Image
1			
2			
3			
4			
5			

Pada tabel 4 menunjukkan perbandingan histogram grayscale pada setiap gambar sebelum dan sesudah penyisipan pesan. Pada histogram *stego image*, jika dilihat secara detail terdapat sedikit perubahan bentuk grafik yang sangat tipis, ini karena dalam proses penyisipan (*embedding*) karakter pesan disisipkan ke dalam bit-bit yang membuat *pixel* ikut mengalami perubahan. Semakin banyak karakter yang sisipkan, grafik pada histogram akan semakin banyak perubahannya. Hal ini menunjukkan bahwa perubahan pada *stego image* menggunakan LSB standar tanpa di modifikasi terlihat cukup signifikan jika menggunakan histogram dengan jumlah karakter sisipan yang banyak. Perubahan ekstensi file juga berpengaruh terhadap perubahan histogram.

3.2. Analisis Kapasitas Penyisipan dan Nilai PSNR

Pengujian terhadap hasil *stego image* dilakukan dengan melihat nilai *Peak-to-Noise Ratio* (PSNR). PSNR diukur dalam satuan *decibel* (db). PSNR digunakan dengan tujuan yaitu untuk mengetahui kualitas citra cover sebelum dan sesudah disisipkan pesan rahasia, dikatakan baik jika citra memiliki nilai PSNR di atas 30db[8]. Rumus persamaan PSNR dapat dilihat pada persamaan (1)[15].

$$PSNR = 10 \log_{10} \left(\frac{MAX_i^2}{MSE} \right) \tag{1}$$

dimana:

- MAX_i = nilai maksimum dari karakter pada teks yang digunakan
- MSE = nilai perhitungan MSE

Nilai (MAX_i) mengacu pada nilai maksimal pada tabel ASCII 8-bit yaitu 255. Sebelum menentukan nilai PSNR, terlebih dahulu harus ditentukan nilai *Mean Square Error* (MSE), kebalikan dari PSNR dimana semakin rendah nilai MSE maka semakin baik[16]. Perhitungan nilai MSE yaitu dengan menggunakan rumus (2).

$$MSE = \frac{1}{M} \sum_{y=0}^M [I(y) - I'(y)]^2 \tag{2}$$

dimana:

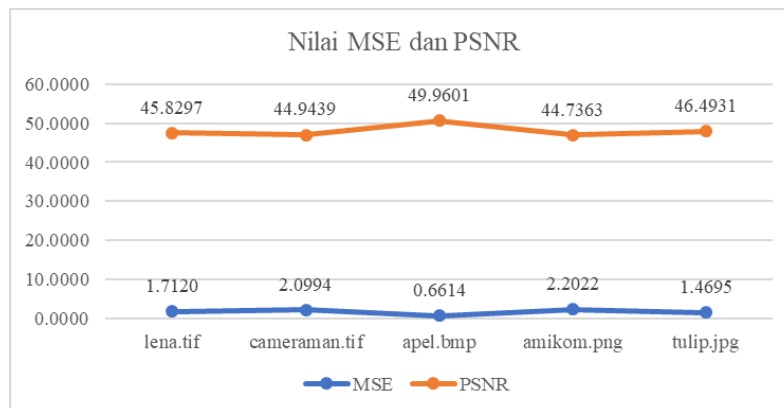
- M = panjang teks stego
- I(y) = nilai desimal karakter dari teks cover
- I'(y) = nilai desimal karakter dari teks stego

Melakukan analisa untuk mengetahui kapasitas penyisipan sangat penting dilakukan. Salah satu pengujian penting dalam steganografi adalah menganalisa kapasitas penyisipan dan nilai PSNR. Kapasitas penyisipan data dilihat ada Tabel 5.

Tabel 5. Nilai Kapasitas

No	Cover Image	Stego Image	Kapasitas Penyisipan
1	lena.tif	lena.bmp	55000 karakter (6875 bit)
2	cameraman.tif	cameraman.bmp	55000 karakter (6875 bit)
3	apel.bmp	apel.bmp	55000 karakter (6875 bit)
4	amikom.png	amikom.bmp	55000 karakter (6875 bit)
5	tulip.jpg	tulip.bmp	55000 karakter (6875 bit)

PSNR memiliki standar nilai pada setiap *cover image* yang disisipkan pesan yaitu $>30\text{db}$ sementara MSE sebaliknya semakin kecil semakin baik. Jika nilai tersebut telah dipenuhi maka gambar tersebut telah memenuhi syarat *imperceptibility*[17]. *Imperceptibility* adalah keberadaan pesan rahasia yang tidak dapat di lihat atau dipersepsi oleh panca indra seperti mata atau telinga. Contohnya jika pesan berupa teks dan disisipkan pada sebuah citra maka penyisipan pesan membuat citra sukar dibedakan atau dilihat oleh mata. Sedangkan kualitas citra yang dikatakan tidak cukup baik berada dibawah 30db . Nilai PSNR pada 5 skenario uji di penelitian ini dapat dilihat pada Gambar 5.



Gambar 5. Hasil Nilai MSE dan PSNR

Pada Gambar 5 keseluruhan skenario uji mendapatkan nilai PSNR $>30\text{db}$ artinya penggunaan LSB dan pengembangan modifikasi yang dilakukan pada pembuatan Aplikasi Keamanan Data dengan Modifikasi Kunci Algoritma Kriptografi untuk Pesan Rahasia Pada Gambar berhasil dan ketahanan pesan yang sisipkan cukup baik dan aman. Nilai PSNR tertinggi yaitu pada file *apel.bmp* ini karena *cover image* tidak mengalami perubahan ekstensi format, nilai MSE 0,66141 dan PSNR 49,9601.

4. Kesimpulan

Berdasarkan dari penelitian dan analisis yang telah dilakukan menghasilkan beberapa temuan dari pengujian Aplikasi Keamanan Data dengan Modifikasi Kunci Algoritma Kriptografi untuk Pesan Rahasia Pada Gambar yaitu penyisipan menggunakan algoritma LSB secara visual tidak terlihat perubahan warna, *noise* dan *pixel* pada gambar. *Stego image* yang dihasilkan memenuhi *imperceptibility*. Dibuktikan dari analisis penyisipan dan nilai PSNR yang menunjukkan nilai PSNR $>40\text{db}$. Perubahan ekstensi format *image* menjadi salah satu faktor dari perubahan grafik histogram dan nilai PSNR yang dihasilkan. Dibuktikan dengan ekstensi gambar (.bmp) memiliki nilai MSE paling rendah dan PSNR paling tinggi yaitu, MSE=0,66141 dan PSNR=49,9601.

Adapun saran untuk penelitian selanjutnya agar dapat ditarik kesimpulan dan temuan yang lebih banyak yaitu melakukan pengujian dengan menganalisis terhadap ketahanan kriptografi dan steganografi dari Aplikasi Keamanan Data dengan Modifikasi Kunci Algoritma Kriptografi untuk Pesan Rahasia Pada Gambar, lakukan lebih banyak skenario uji dengan ukuran *pixel* yang berbeda-beda, lakukan analisis histogram terhadap citra berwarna seperti RGB, lakukan pengujian dengan menggunakan skenario pengiriman gambar melalui sosial media, lakukan perbandingan metode steganografi LSB dengan metode yang lainnya menggunakan analisis histogram.

Daftar Pustaka

- [1] D. Ariyus, *Kriptografi Keamanan Data dan Komunikasi*. Yogyakarta: Graha Ilmu, 2006.
- [2] D. Ariyus, *Pengantar Ilmu Kriptografi (Teori, Analisis dan Implementasi)*. Yogyakarta: ANDI OFFSET, 2008.
- [3] E. Setyaningsih, P. Studi, I. Komputer, and F. S. Terapan, "Penyandian citra menggunakan metode," vol. 2, no. 28, pp. 213–217, 2009.
- [4] D. R. Ignatius, M. Setiadi, C. Jatmoko, E. H. Rachmawanto, C. A. Sari, and M. Subtitusi, "Kombinasi Cipher Subtitusi (Beaufort Dan Vigenere)," pp. 978–979, 2018.
- [5] D. Ariyus, "Optimization Substitution Cipher And Hidden Plaintext In Image Data Using LSB Method," p. 12033, 2019.
- [6] B. Ge, H.-B. Luo, B. Ge, and H.-B. Luo, "Image Encryption Application of Chaotic Sequences Incorporating Quantum Keys," *Int. J. Autom. Comput.*, vol. 16, no. 2, pp. 1–16, Apr. 2019.
- [7] T. W. P. and R. A. N. Danny Adiyani Z., "Implementation of Secure Steganography on Jpeg Image Using LSB Method," *Int. J. Appl. Eng. Res.*, vol. 13, no. 1, pp. 442–448, 2018.
- [8] R. W. Simbolon, "Cipher Dan Steganografi Dengan Teknik Least Significant Bit (Lsb) Protecting the Student Academic Transcript Using Playfair Cipher Cryptography," *J. Teknol. Inf. dan Komun.*, vol. 5, no. 1, pp. 59–70, 2016.
- [9] A. A. Fikhri, "Analisis Perbandingan Histogram dan Kualitas Citra Pada Image Steganografi Menggunakan Metode One Bit Least Significant Bit A-44 A-45," in *Procedia Seminar Nasional Politeknik Negeri Lhokseumawe*, 2018, vol. 2, no. 1, pp. A44-49.
- [10] V. Nagaraj, V. Vijayalakshmi, and G. Zayaraz, "Modulo based Image Steganography Technique against Statistical and Histogram Analysis," 2011.
- [11] S. Jeevitha and N. Amutha Prabha, "A Comprehensive Review On Steganographic Tecniques And Implementation," vol. 13, no. 17, 2018.
- [12] G. Guntoro and M. Fikri, "Perancangan Aplikasi Single Sign-On Menggunakan Autentikasi Gambar," *Digit. Zo. J. Teknol. Inf. dan Komun.*, vol. 9, no. 1, pp. 12–21, 2018.
- [13] D. Suhartono, A. G. Salman, Rojali, and C. Octavianus, "Aplikasi Penyembunyian Pesan Pada Citra Jpeg Dengan Algoritma F5 Dalam Perangkat Mobile Berbasis Android," *Semin. Nas. Apl. Teknol. Inf.*, vol. 2012, no. Snati, pp. 15–16, 2012.
- [14] A. Ardiansyah, "Modifikasi dan Kombinasi Substitusi Cipher Pada Data Citra dengan Metode Least Significant Bit," Universitas AMIKOM Yogyakarta, 2019.
- [15] C. K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," *Hournal Pattern Recognit. Soc.*, vol. 37, no. 3, pp. 469–474, 2004.
- [16] M. K. Achmad Ardiansyah, "Penyembunyian Pesan Rahasia Pada Citra Digital Dengan Teknik Steganografi Menggunakan Metode Least Significant Bit (LSB)," *J. Teknol. Inf.*, vol. XIII, no. November, pp. 96–101, 2018.
- [17] A. Horé and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proceedings - International Conference on Pattern Recognition*, 2010, pp. 2366–2369.

