PERBANDINGAN ALGORITMA *PIXEL VALUE DIFFERENCING* DAN *MODULUS FUNCTION* PADA STEGANOGRAFI UNTUK MENGUKUR KUALITAS CITRA DAN KAPASITAS PENYIMPANAN

Nurhuda Budi Pamungkas[1], Dedi Darwis[2], Ditha Nurjayanti[3], Agung Tri Prastowo[4]

[1,2,3,4]Fakultas Teknik dan Ilmu Komputer, Universitas Teknokrat Indonesia
Jl. ZA. Pagar Alam No. 9-11, Bandar Lampung Indonesia 35132
Telp. (0721) 702022 Fax. (0721) 702022
e-mail : nurhuda.budi@teknokrat.ac.id, darwisdedi@teknokrat.ac.id,
dithanurjayanti@teknokrat.ac.id, agung.tri.prastowo@teknokrat.ac.id

ABSTRACT

Data security and confidentiality is an important aspect of information. The importance of the value of information in every aspect can make it possible to transfer information or data theft by unauthorized parties. One technique for securing data and information is to apply the use of steganography techniques. This study compares the two methods of steganography namely Pixel Value Differencing (PVD) and Modulus Function (MF) to find out which method is better in terms of image quality and message storage capacity. This research was developed using the Python programming language with the images tested in both methods are PNG format images. Based on the results of tests conducted by the PVD algorithm produces better image quality than the MF algorithm because it produces Peak Signal to Noise Ratio (PSNR) values with an average of more than 40dB and the PVD algorithm also produces a stego image that can accommodate more messages than the MF algorithm .

Keywords*: Data Security, MF, PSNR, PVD, Steganography*

ABSTRAK

Keamanan dan kerahasiaan data merupakan salah satu aspek penting dari suatu informasi. Pentingnya nilai informasi pada setiap aspek dapat memungkinkan adanya usaha pemindah alihan atau pencurian informasi atau data oleh pihak yang tidak berwenang. Adapun salah satu teknik untuk mengamankan data dan informasi yaitu dengan menerapkan penggunaan teknik steganografi. Penelitian ini membandingkan dua metode pada steganografi yaitu *Pixel Value Differencing (PVD)* dan *Modulus Function (MF)* untuk mengetahui metode mana yang lebih baik dalam hal kualitas citra dan kapasitas penyimpanan pesan. Penelitian ini dikembangkan menggunakan bahasa pemrograman *Phyton* dengan citra yang diuji pada kedua metode adalah citra berformat *PNG*. Berdasarkan hasil pengujian yang dilakukan algoritma *PVD* menghasilkan kualitas citra yang lebih baik dari algoritma *MF* karena menghasilkan nilai *Peak Signal to Noise Ratio (PSNR)* dengan rata-rata lebih dari 40db dan algoritma *PVD* juga menghasilkan *stego image* yang lebih banyak dapat menampung pesan dibandingkan algoritma *MF*.

Kata Kunci *:* Keamanan Data, *MF, PSNR, PVD,* Steganografi

I. PENDAHULUAN

Masalah keamanan dalam transfer data menjadi perhatian yang paling utama karena pentingnya nilai informasi pada setiap aspek dapat memungkinkan adanya usaha pemindah alihan atau pencurian informasi ataupun data oleh pihak yang tidak berwenang[1,2]. Media penyimpanan dan penyebaran data atau informasi yang digunakan menjadi salah satu alasan rentannya data atau informasi mudah diambil oleh pihak yang tidak bertanggung jawab. Hal ini disebabkan oleh sistem keamanan yang kurang efisien dalam memproteksi kerahasiaan data maupun informasi[3,4,5]. Ada banyak teknik yang dapat digunakan untuk mengamankan informasi diantaranya *encryption, watermarking, digital watermarking, reversible watermarking, cryptography, steganography* dan lain-lain[6]. Pada penelitian ini membahas cara mengamankan data menggunakan steganografi yaitu ilmu yang berhubungan dengan keamanan rahasia data yang tertanam dalam media seperti gambar, teks, audio dan video untuk mempersulit pihak ketiga dalam mendeteksi pesan tetapi dapat memperoleh data rahasia ketika data dikirim melalui saluran publik[7]. Teknik steganografi merekomendasikan beberapa algoritma atau metode yang dapat digunakan untuk mengamankan informasi rahasia. Beberapa metode yang sering digunakan adalah metode *Least Significant Bit (LSB), Discrete Wavelet Transform (DWT), Descrete Cosine Transform (DCT), Pixel Value Differencing* (PVD), *Modulus Function* (MF), dan metode yang lainnya.

Masalah yang umum sering terjadi pada steganografi adalah kualitas citra dan kapasitas penampungan, kedua hal tersebut menjadi tolak ukur dalam menentukan metode steganografi yang baik[8]. Pada penelitian ini membahas perbandingan antara algoritma *Pixel Value Differencing* (PVD) dan *Modulus Function* (MF) agar dapat mengetahui algoritma manakah yang paling baik dalam segi kualitas citra dan kapasitas penyimpanan sehingga dapat dijadikan referensi sebagai bahan pertimbangan ketika akan menggunakan salah satu dari kedua metode tersebut. Penelitian ini bertujuan untuk mengetahui nilai perbandingan kualitas citra dan kapasitas penyimpanan dari metode *PVD* dan *MF*.

II. METODE PENELITIAN

Pada tahap ini, hal yang pertama dilakukan adalah kajian literatur yaitu melalui buku dan jurnal, setelah itu berlanjut pada tahap pengumpulan data yaitu mengumpulkan gambar primer

sebanyak tujuh gambar yang diambil secara langsung menggunakan kamera *Handphone* dan gambar sekunder sebanyak tujuh citra yang didapat dari penelitian sebelumnya sebagai citra pembanding dengan jenis citra yang digunakan berformat *PNG* atau *JPG*[6] yang dijadikan sebagai citra penampung *(cover image)* serta mengidentifikasi permasalahan dan ruang lingkup penelitian. Dengan menggunakan dua algoritma sebagai bahan pembanding kemudian dilakukan sebuah penyisipan pesan ke dalam *cover image* untuk memastikan proses *Embedding* dan *Extraction* dapat berjalan dengan baik. Selanjutnya untuk mengukur kualitas citra maka dilakukan pengujian dengan cara menghitung nilai *Mean Square Error (MSE)* dan *Peak Signal to Noise Rato (PSNR)* menggunakan persamaan (1) dan (2) [9].

$$\mathrm{MSE}_{\mathrm{AVG}} = \frac{\mathrm{MSE_R}+\mathrm{MSE_G}+\mathrm{MSE_B}}{3} \quad (1)$$

$$\mathrm{PSNR} = 10_{\log 10}\left(\frac{255^2}{\mathrm{MSE}}\right) \quad (2)$$

Penelitian ini dikembangkan menggunakan bahasa pemrograman *Python* untuk menguji algoritma mana yang lebih baik dalam hal kualitas citra dan kapasitas penyimpanan pesan antara algoritma *PVD* dan algoritma *MF*.

2.1 Pixel Value Differencing (PVD)

Algoritma *PVD* bekerja pada sepasang piksel bertetangga. Proses penyisipan pesan dilakukan dengan memodifikasi selisih nilai piksel. Salah satu rentang nilai keabuan yang diusulkan oleh *Wu dan Tsai* adalah (8 8 16 32 64 128) dengan jumlah bit n (3 3 4 5 6 7) [10,11]. Tetapi pada penelitian ini dilakukan modifikasi dengan rentang nilai keabuan (8 8 16 32 64 128) dengan jumlah *bit* n (3 3 3 3 3 3). Proses penyisipan algoritma *PVD* dapat dinyatakan dalam langkah-langkah sebagai berikut :

1. Ubah pesan menjadi bilangan *biner* 8 *bit*.
2. Hitung selisih 2 piksel bertetangga (g_i, g_{i+1}) :$(d_i = g_{i+1} - g_1)$
3. Tentukan batas bawah (I_k) dan jumlah *bit* n, dengan cara :
$$I_k \leq d_i < I_{k+1}$$
4. Ambil pesan sebanyak n *bit*, kemudian ubah menjadi desimal (b)
5. Hitung selisih nilai yang baru :
$$d' = \begin{cases} I_k + b, & d \geq 0 \\ -(I_k + b), & d < 0 \end{cases}$$
6. Hitung : $m = d' - d$
7. Hitung nilai piksel baru :
$$f(g'_i, g'_{i+1}) = \begin{cases} \left(g_i - \left[\frac{m}{2}\right], g_{i+1} + \left[\frac{m}{2}\right]\right), & m = ganjil \\ \left(g_i - \left[\frac{m}{2}\right], g_{i+1} + \left[\frac{m}{2}\right]\right), & m = genap \end{cases}$$

Ekstraksi pesan algoritma *PVD* dilakukan dengan cara berikut ini :

1. Hitung selisih piksel bertetangga
$$(g_i, g_{i+1})\ d_i = g_{i+1} - g_i$$

2. Tentukan batas bawah (I_k) dan jumlah *bit* n, dengan cara :

$$I_k \leq d_i < I_{k+1}$$

3. Hitung : $b = |d| - I_k$
4. Ubah ***b*** (desimal) menjadi biner ***n*** *bit*
5. Ambil Pesan = *bit* n.

Urutan proses penyisipan pesan algoritma *PVD* dapat dilihat pada Gambar 1.

Gambar 1. *Flowchart PVD*

2.2 Modulus Function (MF)

Pada algoritma *MF* proses penyisipan pesan dilakukan melalui modifikasi nilai sisa hasil bagi (*remainder*) dari piksel bertetangga [10,12]. Proses penyisipan algoritma *MF* dapat dinyatakan pada langkah-langkah berikut ini :

1. Ubah pesan menjadi bilangan *biner* 8 *bit*.
2. Hitung selisih 2 piksel bertetangga (g_i, g_{i+1}) $d_i = |g_{i+1} - g_i|$
3. Tentukan batas bawah (I_k) dan jumlah bit ***n***, dengan cara :

$$I_k \leq d_i < I_{k+1}$$

4. Ambil pesan sebanyak ***n*** bit, kemudian ubah menjadi desimal (***b***)
5. Hitung nilai sisa hasil bagi (*remainder*)

$$r = (g_i + g_{i+1}) mod\ 2^n$$

6. Hitung :

$$m = |r - b| dan\ m' = |2^n - m|$$

7. Tentukan nilai piksel baru (g_i', g_{i+1}) dengan cara berikut :
   a. Jika $r > b$ dan $m \leq {2^n}/{2}$ dan $g_i \geq g_{i+1}$ maka
   $(g_i', g_{i+1}') = (g_i - \left\lceil \frac{m}{2} \right\rceil, g_{i+1} - \left\lceil \frac{m}{2} \right\rceil)$
   b. Jika $r > b$ dan $m \leq 2^n/2$ dan $g_i < g_{i+1}$ maka
   $(g_i', g_{i+1}') = (g_i - \left\lceil \frac{m}{2} \right\rceil, g_{i+1} - \left\lceil \frac{m}{2} \right\rceil)$
   c. Jika $r > b$ dan $m > 2^n/2$ dan $g_i \geq g_{i+1}$ maka
   $(g_i', g_{i+1}') = (g_i - \left\lceil \frac{m'}{2} \right\rceil, g_{i+1} - \left\lceil \frac{m'}{2} \right\rceil)$
   d. Jika $r > b$ dan $m > 2^n/2$ dan $g_i < g_{i+1}$ maka
   $(g_i', g_{i+1}') = (g_i - \left\lceil \frac{m'}{2} \right\rceil, g_{i+1} - \left\lceil \frac{m'}{2} \right\rceil)$
   e. Jika $r \leq b$ dan $m \leq 2^n/2$ dan $g_i \geq g_{i+1}$ maka
   $(g_i', g_{i+1}') = (g_i - \left\lceil \frac{m}{2} \right\rceil, g_{i+1} - \left\lceil \frac{m}{2} \right\rceil)$
   f. Jika $r \leq b$ dan $m \leq 2^n/2$ dan $g_i < g_{i+1}$ maka
   $(g_i', g_{i+1}') = (g_i - \left\lceil \frac{m}{2} \right\rceil, g_{i+1} - \left\lceil \frac{m}{2} \right\rceil)$
   g. Jika $r \leq b$ dan $m > 2^n/2$ dan $g_i \geq g_{i+1}$ maka
   $(g_i', g_{i+1}') = (g_i - \left\lceil \frac{m'}{2} \right\rceil, g_{i+1} - \left\lceil \frac{m'}{2} \right\rceil)$
   h. Jika $r \leq b$ dan $m > 2^n/2$ dan $g_i < g_{i+1}$ maka

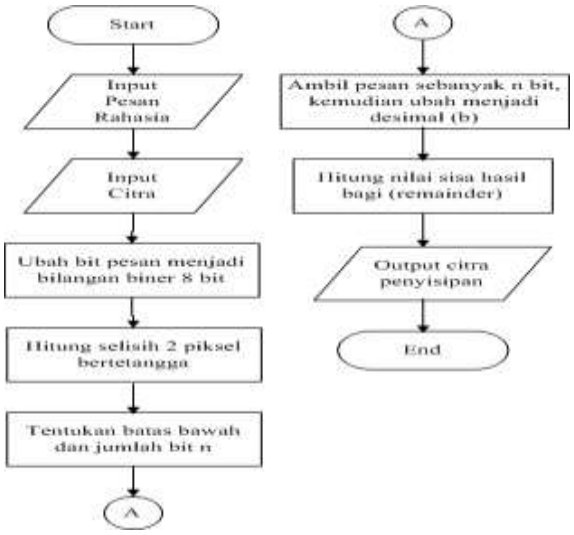
$$(g_i', g_{i+1}') = (g_i - \left[\frac{m'}{2}\right], g_{i+1} - \left[\frac{m'}{2}\right])$$

Sedangkan untuk proses ekstraksi pesan dilakukan dengan cara berikut ini :

1. Hitung selisish 2 piksel bertetangga (g_i, g_{i+1})
   $d_i = |g_{i+1} - g_i|$
2. Tentukan batas bawah (I_k) dan jumlah *bit* ***n***, dengan cara :
   $I_k \leq d_i < I_{k+1}$
3. Hitung nilai sisa hasil bagi (*remainder*)

$$b = (g_i, g_{i+1}) \; mod \; 2^n$$

Proses penyisipan pesan menggunakan algoritma *MF* dapat dilihat pada Gambar 2.



Gambar 2. *Flowchart MF*

III. HASIL DAN PEMBAHASAN

Untuk mengukur kualitas citra dan kapasitas penyimpianan pada steganografi maka metode pengujian yang dilakukan adalah *fidelity* yaitu dengan menguji perbandingan *cover image* dan *stego image*, apakah citra *cover image* tidak jauh berubah setelah terjadi penyisipan pesan dan apakah *stego image* masih terlihat baik setelah disisipi pesan[13]. Pengujian *fidelity* dilakukan dengan cara menghitung nilai *MSE* dan *PSNR*. [9]

3.1 Hasil Pengujjian Fidelity PVD

Hasil Pengujian *fidelity* pada algoritma *PVD* dapat dilihat pada Tabel 1 dan Tabel 2.

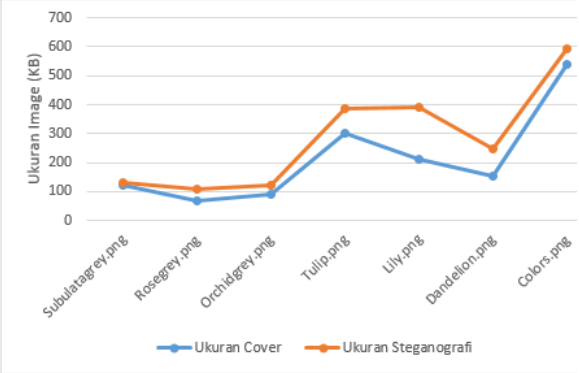
Tabel 1. Hasil Pengujian Algoritma PVD (dengan ukuran pesan yang sama dan *cover image* sama)

Gambar *Cover*	**Ukuran Gambar *Stego* (kb, *bytes*)**	**MSE**	**PSNR**
 Subulatgrey.png	130, 133.852	9.523 db	37.960 db
 *Rosegrey.png*	107, 109.744	6.976 db	39.240 db
 *Orchidgrey.png*	121, 123.959	15.65 4 db	36.185 db
 *Tulip.png*	385, 394.839	5.786 db	40.507 db
 *Lily.png*	391, 401.361	1.119 db	47.644 db
 *Dandelion.png*	246, 252.672	0.594 db	50. 391 db
 *Colors.png*	593, 607.603	100.3 64 db	28.115 db

db = desibel.

Tabel 1 menunjukan pengujian *fidelity* dengan ukuran dimensi yang sama dan pesan yang sama. Pesan yang disisipkan adalah sebuah *file* bernama *message.txt* dengan ukuran sebesar 22,2 KB atau 22.820 *bytes*, memiliki dimensi

512 x 512 piksel. Dari hasil *stego image* dapat disimpulkan bahwa ukuran *stego image* bertambah menjadi lebih besar dari gambar asli atau *cover*. Bertambahnya ukuran *stego* dikarenakan pada tahap penyisipan pada gambar *cover* dalam setiap *pixel* dapat menampung minimal 3 *bit* pesan dan setiap *bit* disisipkan dalam *chanel* R, G, dan B.



Gambar 3. Grafik Hasil Pengujian *Fidelity* untuk *PVD*

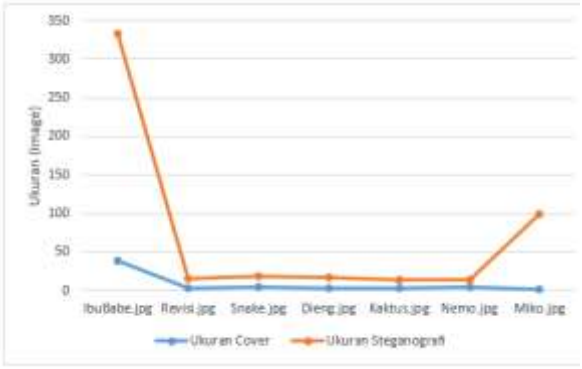
Gambar 3 menunjukan bahwa ukuran *file stego image* bertambah dibandingkan dengan ukuran *cover image* atau gambar asli sebelum disisipkan pesan. Bertambahnya ukuran *stego* dikarenakan pada tahap penyisipan dilakukan iterasi sebanyak *bit* n, sedangkan dalam setiap iterasi menghasilkan perubahan nilai piksel yang signifikan.

Tabel 2. Hasil Pengujian Algoritma PVD (dengan ukuran pesan yang sama dan *cover image* berbeda)

Gambar *Cover*	**Dimensi Gambar *Cover***	**Ukuran Gambar *Stego* (kb, *bytes*)**	**MSE**	**PSNR**
 IbuBabe.jpg	384 x 512	331, 339.623	5.638 db	40.620 db
 Revisi.jpg	3096 x 4128	14,9, 15.680.857	0.018 db	65.593 db
 Snake.jpg	5184 x 3456	19,0, 19.931.592	0.017 db	65.874 db
 Dieng.jpg	5152 x 3864	16,2, 17.086.125	0.012 db	67.357
 Kaktus.jpg	3264 x 2448	13,3, 14.015.984	0.269 db	53.834 db
 Nemo.jpg	4000 x 3000	13,9, 14.593.817	0.018 db	65.679 db
 Miko.jpg	3264 x 2448	9,89, 10.371.747	0.024 db	64.375 db

db = desibel.

Tabel 2 menunjukan hasil *stego image* menggunakan gambar *cover* yang berbeda dan ukuran dimensi yang berbeda namun pesan yang sama yaitu 22,2 KB. Ukuran *stego image* menjadi lebih kecil dibandingkan gambar asli atau *cover*. mengecilnya ukuran *stego image* dikarenakan proses penyisipan *bit* pesan pada gambar *cover*.



Gambar 4. Grafik Hasil Pengujian *Fidelity* untuk *PVD* (dengan ukuran pesan yang sama dan *cover image* berbeda)

Pada Gambar 4 dengan pengujian menggunakan citra yang berbeda dan pesan yang sama menunjukkan hasil yang sama bahwa ukuran *stego image* bertambah. Ukuran *stego image* yang bertambah disesabkan karena adanya perubahan nilai piksel.

Berdasarkan hasil perbandingan dari citra rgb dan *grayscale* dari Tabel 1 dan Tabel 2 pengujian *fidelity* dengan menggunakan algoritma *PVD* menunjukan nilai *PSNR* antara gambar *cover* dan *stego image* kurang baik karena terdapat beberapa gambar *cover* yang masih memiliki nilai PSNR dibawah 40db dan terdapat beberapa gambar yang memiliki nilai MSE diatas 1,0 db. artinya kualitas gambar *cover* mengalami perubahan yang signifikan.

3.2 Hasil Pengujian Fidelity MF

Hasil Pengujian *fidelity* pada algoritma *MF* dapat dilihat pada Tabel 3 dan Tabel 4.

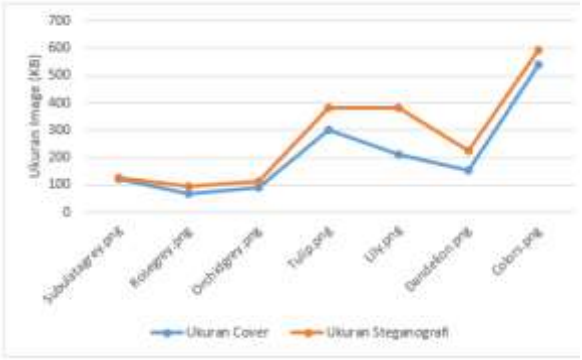
Tabel 3. Hasil Pengujian Algoritma *MF* (dengan ukuran pesan yang sama dan *cover image* sama)

Gambar *Cover*	**Ukuran Gambar *Stego* (kb, *bytes*)**	**MSE**	**PSNR**
 Subulatgrey.png	123, 126.335	0.694 db	49.335 db
 *Rosegrey.png*	94,7, 97.000	21.848 db	34.282 db
 *Orchidgrey.png*	110, 112.951	6.373 db	40.088 db
 *Tulip.png*	382, 391.185	6.381 db	40.082 db
 *Lily.png*	382, 391.208	7.204 db	39.555 db
 *Dandelion.png*	226, 232.354	0.232 db	54.482 db
 *Colors.png*	608, 623.279	10.068 db	38.101 db

db = desibel

Tabel 3 menunjukan pengujian tehadap gambar *cover* yang sama dan dan ukuran pesan yang sama. Pesan yang disisipkan adalah sebuah *file* bernama *message.txt* dengan ukuran sebesar 22,2 KB atau 22.820 *bytes*, memiliki dimensi 512 x 512 piksel. Dari hasil *stego image* dapat disimpulkan bahwa ukuran *stego image* bertambah menjadi lebih besar dari gambar asli atau *cover*. Bertambahnya

ukuran *stego* dikarenakan pada tahap penyisipan pada gambar *cover* dalam setiap *pixel* dapat menampung minimal 3 *bit* pesan dan setiap bit disisipkan dalam *chanel* R, G, dan B.



Gambar 5. Grafik Hasil Pengujian *Fidelity* untuk *MF* (dengan ukuran pesan yang sama dan *cover image* sama)

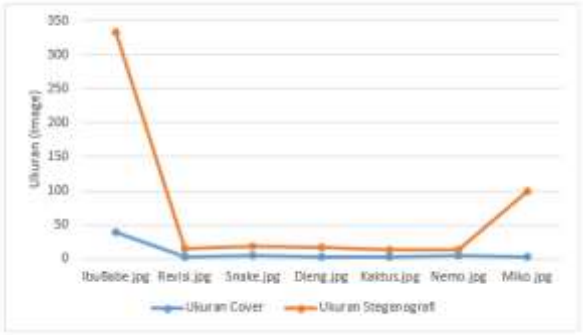
Grafik Gambar 5 dan Tabel 3 menunjukan perubahan pada ukuran *stego image*, namun perubahan pada *stego image Subulatagrey.png* tidak mengalami perubahan yang terlalu besar. Tetapi jika dibandingkan dari segi kualitasnya, *stego image Dandelion.png* jauh lebih baik kualitasnya dengan perbandingan nilai *MSE* untuk *stego image Lily.png* sebesar 0.623db dan *Dandelion.png* sebesar 0.232 db.

Tabel 4. Hasil Pengujian Algoritma *MF* (dengan ukuran pesan yang sama dan *cover image* berbeda)

Gambar *Cover*	**Dimensi *Cover***	**Ukuran Gambar *Stego* (kb, *bytes*)**	**MSE**	**PSNR**
 IbuBabe.jpg	384 x 512	332, 340.436	0.307 db	53.261 db
 Revisi.jpg	3096 x 4128	14,9, 15.669.927	0.005 db	71.349 db
 Snake.jpg	5184 x 3456	18,9, 19.920.765	0.090 db	58.602 db
 Dieng.jpg	5152 x 3864	22,2, 22.820	0.003 db	73.291 db
 Kaktus.jpg	3264 x 2448	22,2, 22.820	0.145 db	56.524 db
 Nemo.jpg	4000 x 3000	22,2, 22.820	0.528 db	50.903 db
 Miko.jpg	3264 x 2448	22,2, 22.820	0.974 db	48.247 db

db = desibel

Tabel 4 menunjukan hasil *stego image* menggunakan gambar *cover* yang berbeda dan ukuran dimensi yang berbeda namun pesan yang sama (22,2 KB). Ukuran *stego image* menjadi lebih besar dibandingkan gambar asli atau *cover*. Bertambah besarnya ukuran *stego image* dikarenakan proses penyisipan *bit* pesan pada gambar *cover*.



Gambar 6. Grafik Hasil Pengujian *Fidelity* untuk *MF* (dengan ukuran pesan yang sama dan *cover image* berbeda)

Pada Gambar 5 grafik dengan pengujian menggunakan citra yang berbeda dan pesan yang sama menunjukkan hasil yang sama bahwa ukuran *stego image* bertambah. Ukuran *stego image* yang bertambah disesabkan karena adanya perubahan nilai piksel.

Berdasarkan hasil perbandingan dari citra RGB dan *grayscale* dari Tabel 3 dan Tabel 4 pengujian dengan menggunakan algoritma *MF* menunjukan nilai *PSNR* antara gambar *cover* dan *stego image* memiliki kualitas yang cukup baik dengan nilai *PSNR* rata-rata lebih dari 40,0db tetapi hampir keseluruhan memiliki nilai *MSE* rata-rata lebih dari 6,000 db dan untuk beberapa pesan yang ukurannya kecil dapat dinyatakan memiliki hasil kualitas yang baik karena nilai *PSNR* lebih dari 40db dan *MSE* dibawah 0,145db. Algoritma steganografi yang baik berdasarkan kualitas citra adalah yang memiliki nilai PSNR lebih dari 40db dan memiliki nilai MSE yang semakin mendekati angka 0 atau semakin kecil[14].

IV. SIMPULAN

Berdasarkan hasil pengujian yang dilakukan hasil perbandingan kualitas citra (*fidelity*) menunjukan bahwa *stego image* yang dihasilkan algoritma *PVD* memiliki kualitas citra yang lebih baik dibandingkan dengan *stego image* yang dihasilkan oleh algoritma *MF*. Hal ini dapat dilihat dari nilai rata-rata *PSNR* yang dihasilkan oleh algoritma *PVD* lebih besar dibandingkan nilai rata-rata *PSNR* yang dihasilkan oleh algoritma *MF*. Begitu juga untuk kapasitas penyimpanan algoritma *PVD* memiliki kapasitas yang lebih baik dibandingkan dengan algoritma *MF* karena algoritma *PVD* dapat menampung pesan dengan ukuran yang besar dibandingkan algoritma *MF*.

UCAPAN TERIMA KASIH

Ucapan terimakasih kami sampaikan kepada Kementerian Riset dan Teknologi / Badan Inovasi Nasional yang telah memberikan hibah Penelitian Dosen Pemula (PDP) tahun pelaksanaan 2020.

DAFTAR PUSTAKA

[1] Jonathan, I., Haryono, A.Y., Leonardi, K., 2017. Penelitian Mengenai Metode Steganografi

Least Significant Bit. *ULTIMA Computing*, Volume IX, hal. 17-20.

[2] Suhendri, A., Juniansyah, B.D., Priono, M.J., Darwis, D., Implementasi Kombinasi Affine Cipher dan One Time Pad dalam Pengamanan Pengiriman Pesan. *Jurnal Informatika*, Vol.18 No.2, hal. 124-129.

[3] Zhao, J. X. Q., 2015. Data Embendding Based on Pixel Value Differencing and Modulus Function using Indeterminate Equation. *The Journal of China Universitas of Posts and Telecommunications,* Issue 22(1), hal. 95-100.

[4] Li, H., 2018. Steganography With Pixel Value Differencing and Modulus Function Based on PSO. *Journal of Information Security and Applications,* Issue 43, hal. 47-52.

[5] Mishra, M., Mishra, P., 2012. Digital *Image* Data Hiding Techniques, *ANSVESA*, Volume 7 No.2, hal. 105–115.

[6] Darwis, D., Junaidi, A., Wamiliana., 2019. A New Approach of Steganography Using Center Sequential Technique, *Journal of Physics: Conference Series*, Volume 1338. No. 1, hal.1-6

[7] Darwis, D., Prabowo, R., Hotimah, N., 2018. Kombinasi Gifshuffle, Enkripsi AES dan Kompresi Data Huffman Untuk Meningkatkan Keamanan Data. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK),* Volume 5, No.4, hal. 389-394.

[8] Awate., Patil, M.M., 2016. Modulus Function and Pixel Value Differencing Coupled with Modified Pixel Indicator Based Secret Data Hidding Method. *International Journal of Advances in Science Engineering and Technology,*Vol.4. No.2. hal. 26-29.

[9] Darwis, D., 2016. Implementasi Teknik Steganografi *Least Significant Bit* (LSB) dan Kompresi untuk Pengamanan Data Pengiriman Surat Elektronik. *TEKNOINFO,* Volume 10. No.2, hal. 32-38.

[10] Andono, P. N., Sutojo, T., 2017. *Pengolahan Citra Digital*. Yogyakarta: ANDI.

[11] Pan, L. Y., 2011. *image Steganography* Method Based on PVD and *Modulus Function*. *IEEE,* hal. 282-284.

[12] Tyagi, R. C., 2015. High Capacity Image Steganography Based on Pixel Value Differencing and Pixel Value Sum. *IEEE,* Issue 92, hal. 488-493.

[13] Zulfansyuri, M., 2016. Kombinasi Algoritma *Pixel Value Differencing* Dengan Algoritma Caesar Cipher Pada Proses Steganografi. *Journal Of Computer Engineering, System And Sciences*, Volume I, hal. 19-25.

[14] Mahmood, T, Mehmood, Z, Shah, M., 2018, A robust technique for copy-move forgery detection and localization in digital images via stationary wavelet and discrete cosine transform. *J Vis Commun Image* Volume 53, hal. 202–214.

[15] Fitria, Y. A. (2019). *Visualization of Data on Earthquake Prone Areasfrom the Analysis of Earthquake Data Vibrations*. Test Engineering & Management, 5301-5308.