**Perancangan Steganografi Penyisipan Pesan Teks pada Citra Digital Menggunakan *Pixel Value Differencing***

***Steganography Design of Text-Message Insertion in Digital Images Using Pixel Value Differencing***

David Saputra1, Arita Witanti2

Program Studi Informatika, Fakultas Teknologi Informasi, Universitas Mercu Buana Yogyakarta, Jl. Wates Km. 10 Yogyakarta 55753, Indonesia

Email: [dvdsaputra24@gmail.com](mailto:dvdsaputra24@gmail.com), [arita@mercubuana-yogya.ac.id](mailto:arita@mercubuana-yogya.ac.id)

# ABSTRAK

Keamanan pesan pada era modern ini sangat dibutuhkan. Perkembangan jaringan internet yang memungkinkan orang untuk saling bertukar data melalui jaringan internet. Seiring dengan perkembangan internet, kejahatan teknologi komunikasi dan informasi juga turut berkembang salah satunya adalah pencurian data*.* Masalah tersebut dapat ditangani dengan menggunakan ilmu-ilmu yang sudah ada sejak jaman dahulu yaitu steganografi. Steganografi merupakan teknik yang dapat mengamankan pesan tersebut dengan cara menyembunyikan pesan yang disisipkan melalui media penampung berupa citra digital. *Pixel Value Differencing* sebagai salah satu dari teknik steganografi yang cukup aman dalam menyembunyikan pesan rahasia. *Pixel Value Differencing* adalah metode steganografi yang menggunakan nilai perbedaan antara dua piksel untuk menentukan berapa banyak bit rahasia bisa disisipkan. Berdasarkan hasil pengujian didapat kesimpulan bahwa dengan menggunakan metode *pixel value differencing* (PVD), kapasitas citra untuk menyisipkan pesan, lebih kecil dari ukuran citranya, waktu proses pada metode ini cukup cepat, kualitas citra setelah disisipkan pesan mempunyai kualitas yang baik

**Kata kunci**: *Steganografi, Penyisipan pesan teks, PVD (Pixel Value Differencing).*

ABSTRACT

*In this modern era, message security is needed. The development of the internet network allows people to exchange data through the internet. Along with the internet development, crimes of communication and information technology are also developing, one of which is data stealing. These problems can be solved using science that has existed since the ancient times that is steganography. Steganography is a technique that can secure a message by hiding messages through inserting them into a media in the form of digital images. Pixel Value Differencing as one of steganography techniques is quite safe in hiding secret message. Pixel Value Differencing is a steganographic method that uses values of difference between two pixels to determine how many secret bits that can be inserted. Based on the testing result, it is concluded that using pixel value differencing (PVD) method, the capacity of the image to insert a message is smaller than the image size; the processing time for this method is quite fast; and the image quality after being inserted the message has a good quality.*

**Keywords**: *Steganography, text-message insertion, PVD (Pixel Value Differencing).*

# 1. PENDAHULUAN

Teknologi komunikasi dan informasi berkembang dengan pesat dan memberikan pengaruh besar bagi kehidupan manusia. Sebagai contoh, perkembangan jaringan internet yang memungkinkan orang untuk saling bertukar data melalui jaringan internet. Seiring dengan berkembangan internet, kejahatan teknologi komunikasi dan informasi juga turut berkembang, seperti *hacker* dan *cracker*. Keamanan pesan pada era modern ini sangat dibutuhkan. Untuk memenuhi itu dikembangkan metode–metode baru dalam pengamanan data. Masalah tersebut dapat ditangani dengan menggunakan ilmu-ilmu yang sudah ada sejak jaman dahulu, seperti kriptografi dan steganografi. Seiring berkembangnya ilmu pengetahuan dalam bidang teknologi membuat rentan data yang bersifat rahasia disalah gunakan oleh pihak yang tidak bertanggung jawab. Berbagai solusi diciptakan untuk menanggulangi masalah tersebut, salah satunya yaitu dengan menggunakan teknik steganografi. Steganografi adalah teknik untuk menyembunyikan data rahasia ke dalam wadah (media) digital sehingga keberadaan data rahasia tersebut tidak diketahui oleh orang lain**.**

Steganografi telah berkembang dan memiliki banyak metode, salah satunya adalah *Pixel Value Differencing* (PVD). Metode PVD ini menggunakan selisih nilai piksel yang satu dengan nilai piksel yang lain, dimana hasil selisih kedua piksel tersebut nantinya akan digunakan untuk menyisipkan pesan pada media lain yang ingin disembunyikan. Setelah pesan sudah disisipkan pada media penampung pesan, maka nilai piksel-piksel tersebut akan berubah. Nilainya akan berubah sesuai dengan rumus perhitungan PVD.

# 2. TINJAUAN PUSTAKA

Penelitian dengan judul “Perancangan Steganografi *Hidden Message* dengan Metode *Least Significant Bit Insertion* (LSB) Berbasis Matlab”. Steganografi *(steganography)* adalah ilmu teknik atau seni untuk menyembunyikan pesan rahasia di dalam pesan lain sehingga keberadaan pesan rahasia tersebut tidak dapat diakses oleh orang lain yang tidak mempunyai kewenangan. Dalam keamanan data (teks atau audio) mengikuti 5 (lima) kaidah utama yaitu faktor *confidentiality, integrity, availability, authenticity,* dan *nonrepudiation.* Metode algoritma LSB merupakan metode yang digunakan steganografi dimana proses penggabungan pesan yang berisi teks disimpan dalam file tertentu dalam hal ini format yang dilakukan penelitian ini JPG dan BMP dengan ukuran piksel tertentu. (Nizirwan, 2018)

Penelitian dengan judul “Pengamanan *File* Citra Digital dengan Menggunakan Metode *Least Significant Bit* dan *End Of File*”. Metode penyisipan LSB (Least Significant Bit) menyisipi pesan dengan cara mengganti bit ke 8, 16 dan 24 pada representasi biner *file* gambar dengan representasi biner pesan rahasia yang akan disembunyikan. Teknik EOF atau *End Of* *File* menggunakan cara dengan menyisipkan data pada *file*. Teknik EOF digunakan untuk menyisipkan data yang ukurannya sesuai dengan kebutuhan. Ukuran *file* yang telah disisipkan data sama dengan ukuran *file* sebelum disisipkan data ditambah dengan ukuran data yang disisipkan data kedalam *file* tersebut. (Sinaga & Sitorus, 2017)

# 3. METODOLOGI PENELITIAN

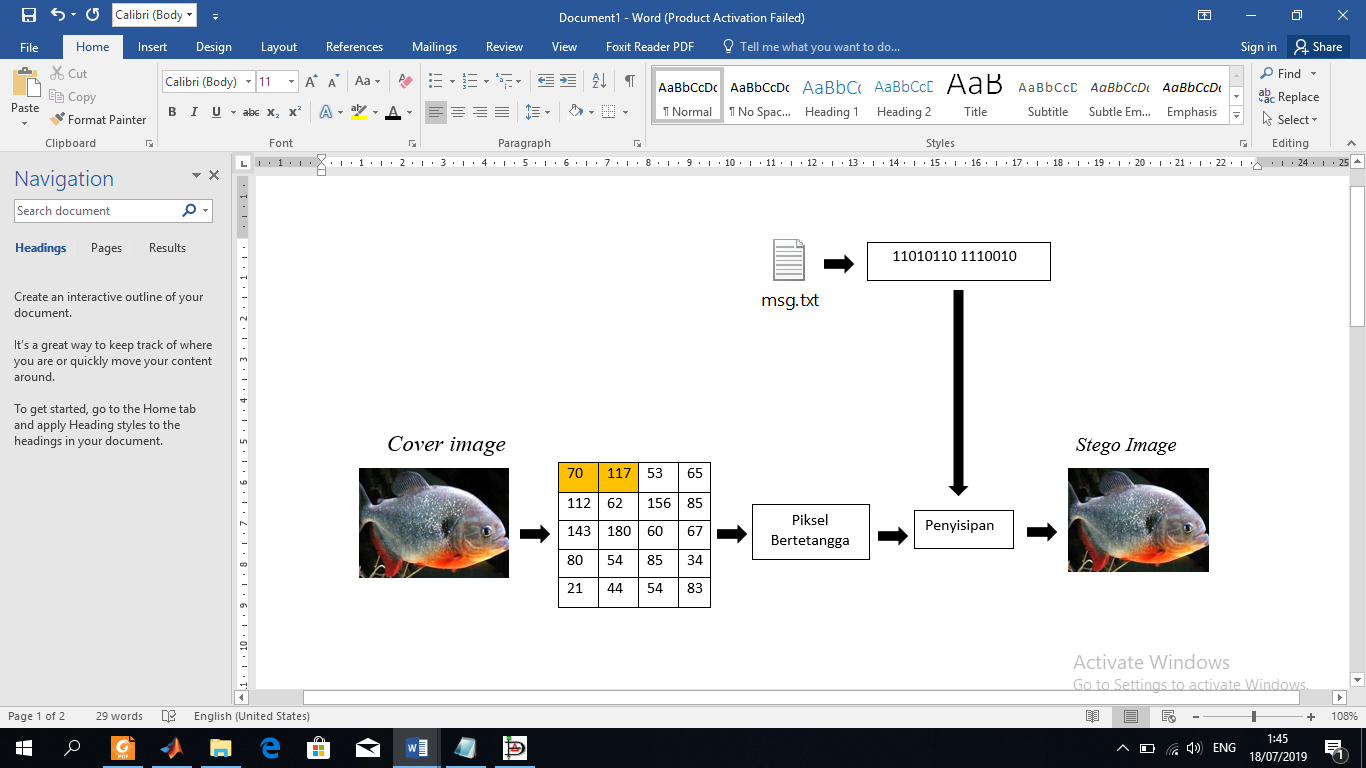
## **Algoritma Penyisipan Pesan**

Pada teknik steganografi PVD proses penyisipan pesan dilakukan dengan cara membagi-bagi piksel pada citra *cover* menjadi pasangan-pasangan yang saling berdekatan dan tidak saling menumpuk. Jumlah bit pesan yang disisipkan bergantung pada selisih setiap pasang piksel tersebut. Sebuah tabel jangkauan digunakan untuk menentukan banyaknya bit yang akan disisipkan. Berikut contoh tabel rentang bit yang dapat disisipkan pada tiap daerah rentang dapat dilihat pada Tabel 1.

Tabel 1. Rentang Bit

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Rentang | 1 | 2 | 3 | 4 | 5 | 6 |
| *Range* | 0-7 | 8-15 | 16-31 | 32-63 | 64-127 | 128-255 |
| *tᵢ* (jumlah bit yang disisipkan) | 3 | 3 | 4 | 5 | 6 | 7 |

Jika selisih pasangan dua piksel bernilai antara 0-7 maka penyisipan berada pada rentang R1 , sehingga aplikasi akan menyisipkan 3 bit pesan kedalam bit *cover.* Selisih antara 8-15 berada pada rentang R2, pesan yang disisipkan sebesar 3 bit. Pada R1 dan R2 ditetapkan bit pesan yang dapat disisipkan sebesar 3 bit. Pada rentang R3 dengan selisih piksel antara 16-31 dapat disisipkan pesan sebesar 4 bit. Rentang R4 dengan nilai selisih antara 32 dan 63 pesan yang disisipkan sebesar 5 bit, rentang R5 sebesar dengan selisih antara 64 dan 127 pesan disisipkan sebesar 6 bit. Pada rentang terakhir, rentang 6 dengan selisih piksel sebesar 128 sampai 255 maka pesan yang disisipkan sebesar 7 bit. Berikut adalah proses penyisipan pesan yang dilakukan dengan menggunakan metode *Pixel Value Differencing* dapat dilihat pada Gambar 1.



Gambar 1. Gambar Penyisipan Pesan

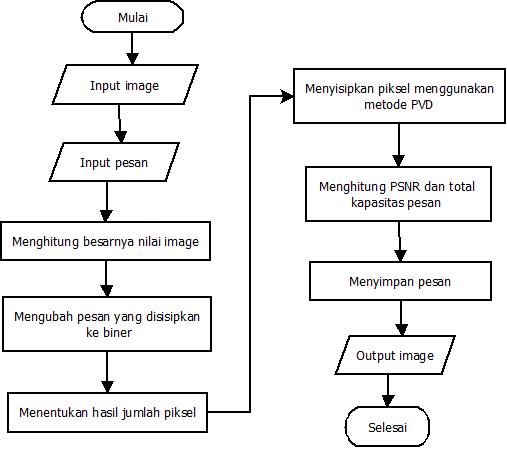
Keterangan Gambar 1:

1. *Cover* citra : Pesan yang digunakan untuk menyembunyikan embedded message.

2. *Stego Image* : Pesan yang sudah disisipkan pesan *embedded message*.

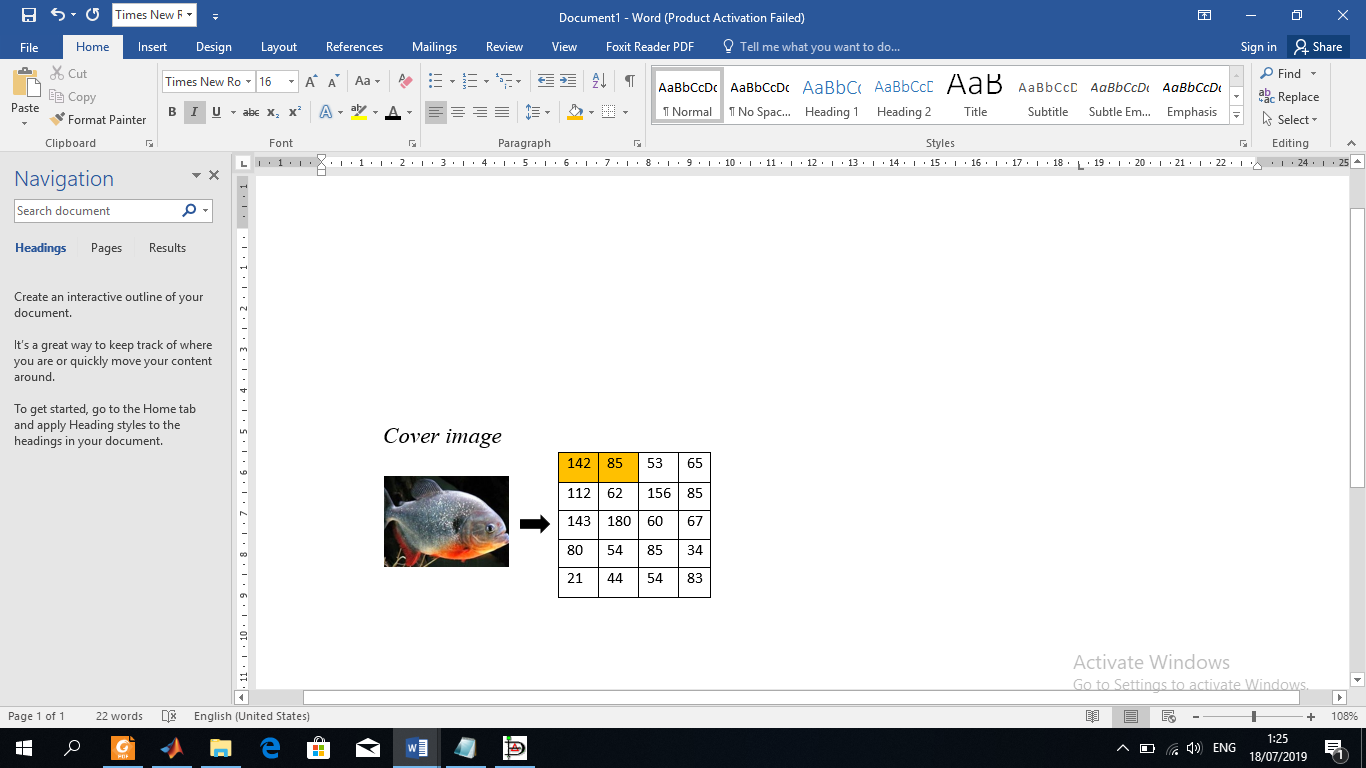
3. *Embedded* : Pesan yang disembunyikan.

Algoritma yang digunakan untuk pengembangan menu penyisipan pesan dapat dilihat pada diagram alu­­­­­r Gambar 2.



Gambar 2. Penyisipan Pesan

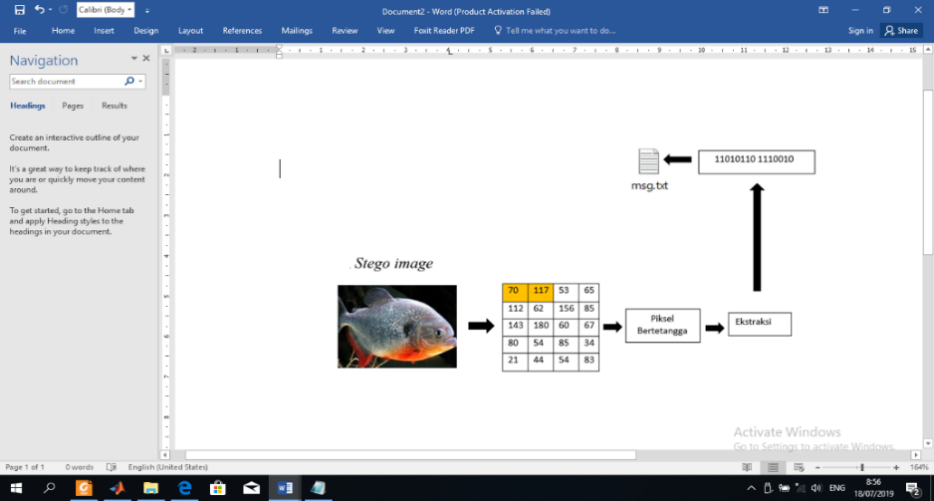
Sistem akan menghitung selisih piksel yang bertetangga dan mengambil nilai piksel dari suatu citra didapat nilai sebagai berikut:



Gambar 3. Nilai piksel Penyisipan

1. *d=Pᵢ-Pᵢ₊₁=* 142−85= 67. *Continues range* dari nilai *difference value* pada *skema wu* dan *tsai* R = {[0,7], [8,15], [16,31], [32,63], [64,127], [128,255]}. Letak *continues range* *dᵢ*= 67 berada dalam rentang [64,127], *bb*=64 dan *ba*=127
2. *t*= Log₂ wᵢ, *t* = Log₂ (127–64) sehingga didapat = 4 bit, maka ambil bit dari pesan sebanyak *t* yaitu 1011. b=11
3. Nilai *difference* *d’ᵢ=bb+b, d’ᵢ*=64+11=75
4. *m=d’ᵢ-dᵢ, m*=85-67=16
5. Aturan terpenuhi *Pᵢ≥Pᵢ₊₁* dan *d’ᵢ>dᵢ,* maka *(Pᵢ+|m/2|,Pᵢ₊₁-|m/2|)* = (142+|12/2|,75-|12/2|)
6. Nilai piksel baru *Pᵢ*=148 dan *Pᵢ₊₁=*69, tahapan ini dilakukan sampai semua pesan tersisipi.

## **Algoritma Ekstraksi Pesan**

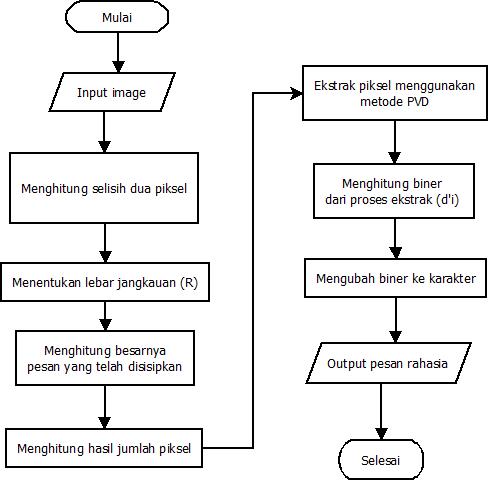
Pengambilan pesan diawali dengan menginput *stego image*. Kemudian dari setiap *layerstego image* dua piksel yang berdekatan *(gₓ, gₓ₊₁)* dihitung selisihnya masing-masing disimpan dalam 3 variabel yang mewakili *layer red, green* dan *blue.* Nilai selisih tersebut kemudian dimutlakkan *(absolute value).* Seperti pada proses penyisipan pesan, demikian pula pada pengambilan pesan digunakan juga tabel jangkauan yang memiliki lima daerah rentang yang sama. Pada setiap iterasi yang menghitung selisih dua piksel berdekatan pada *stego image,* rentang-rentang pada tabel jangkauan digunakan untuk menentukan besarnya bit pesan yang telah disisipkan pada dua piksel stego image. Perhitungan *tistego* dimulai dengan menghitung wⅉ yaitu mengurangkan batas atas dengan batas bawah rentang *(uj-lj₊₁).* Kemudian dilakukan operasi log terhadap *wj* yang hasilnya merupakan besarnya *tistego* maka hasil operasi log terhadap *wj* ditetapkan jika selisih piksel 0-7 dan 8-15 dapat disisipkan 3 bit pesan, selisih 16-31 disisipkan 4 bit, selisih 32-63 disisipkan 5 bit, selisih 64-127 disisipkan 6 bit dan selisih 128-255 disisipkan 7 bit. Pada proses pengambilan pesan, rancangan aliran data dimulai dengan menginput data *stego image* kemudian dilakukan proses ekstraksi dengan menghasilkan *output* data berupa pesan rahasia yang disembunyikan pada *cover.*Proses ini akan menghasilkan informasi yang disembunyikan, dengan masukan berupa citra *stego* objek. Proses ekstraksi pada metode *pixel* *value differencing* dapat dilihat pada Gambar 4.

Gambar 4. Proses Ekstraksi Pesan

Keterangan gambar 4:

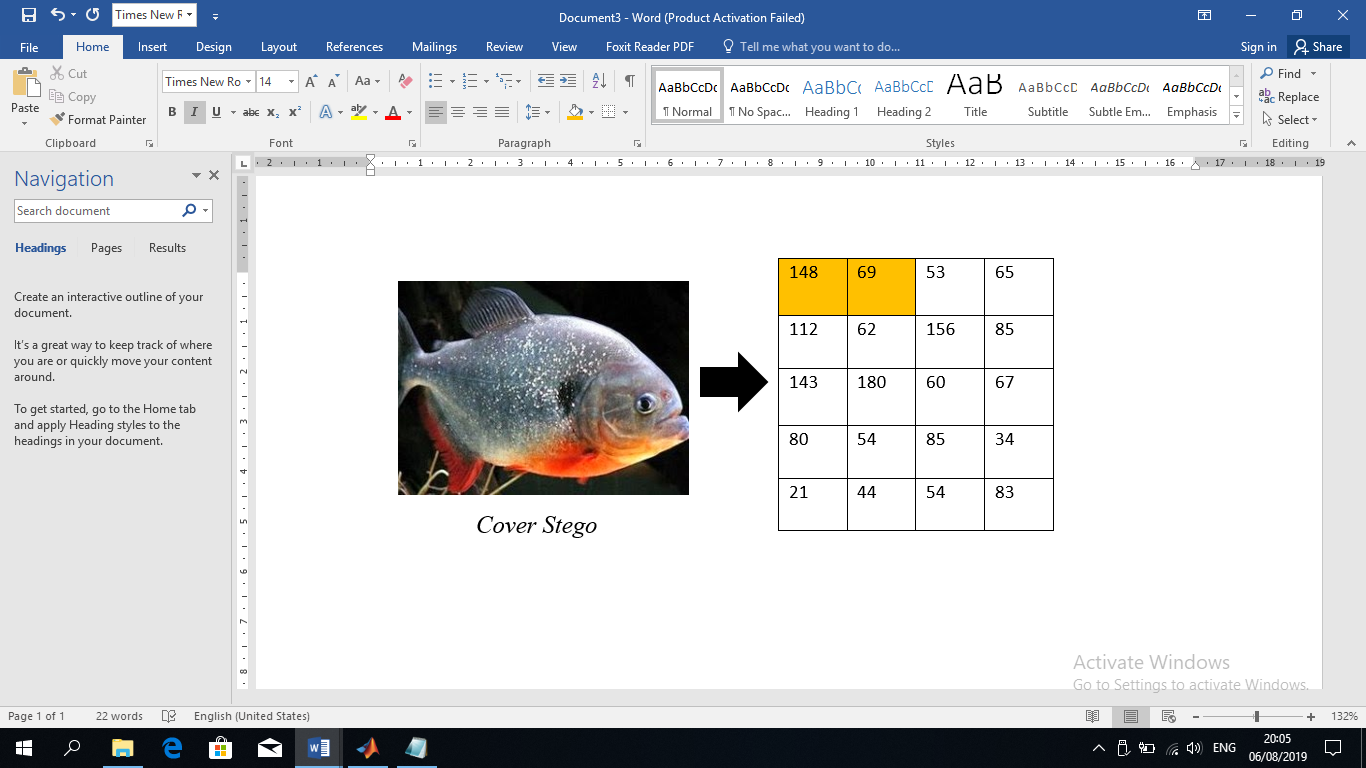
1. Citra *stego*: Pesan yang berisi pesan Embedded message.

2. Ekstraksi pesan: Pengambilan pesan dari citra stego yang sudah disisipi pesan rahasia kedalam pesan asli. Algoritma yang digunakan untuk ekstraksi pesan dapat dilihat pada diagram alu­­­­­r Gambar 5.

****

Gambar 5. Ekstraksi Pesan

Diketahui citra yang digunakan seperti pada Gambar 6.



Gambar 6. Nilai Piksel *Stego*

Proses ekstraksi menggunakan metode *pixel value differencing* dengan tahapan-tahapan sebagai berikut:

1. *Pᵢ* dan *Pᵢ₊₁* adalah piksel yang bertetangga, maka Pᵢ=148 dan Pᵢ₊₁=69.
2. *dᵢ=Pᵢ–Pᵢ₊₁, dᵢ*=148−69=59. *Continues range* dari nilai *difference value* pada skema *wu* dan *tsai* R={[0,7], [8,15], [16,31], [32,63], [64,127], [128,255]}. 59 berada dalam rentang [32, 63], *bb*=32 dan *ba*=63.
3. Menghitung banyak bit dari *image* yang disisipkan kedalam kedua piksel *t=* Log*₂ wᵢ*, *t*=Log2(63–32) sehingga *t*=3, atau terdapat 3 bit pesan yang disisipkan pada kedua piksel.
4. Terakhir mengubah nilai desimal pesan kedalam bentuk bit sebanyak *t*, maka didapat bit pesan *b*=110 dalam desimal 6. Proses berikutnya dilakukan berulang sampai semua piksel diketahui.

## **PSNR**

PSNR *(Peak Signal to Noise Ratio)* digunakan untuk menentukan kualitas gambar setelah disisipi pesan. Gambar *stego* dibandingkan dengan gambar asli untuk menentukan kuali­­tas gambar. Semakin besar nilai PSNR berarti penyisipan pesan ke dalam gambar asli tidak menyebabkan penurunan kualitas gambar *stego.* Sebaliknya jika nilai PSNR semakin kecil maka pada gambar stego akan terjadi penurunan kualitas gambar. Nilai PSNR biasanya mempunyai rentang nilai antara 20 dB sampai dengan 60 dB.

Tabel 2. memperlihatkan kualitas nilai PSNR.

Tabel 2. Nilai PSNR

|  |  |
| --- | --- |
| **Rasio (dB)** | **Kualitas Gambar** |
| 60 dB | *Excellent,* tanpa derau |
| 50 dB | *Good,* terdapat banyak derau tapi kualitas citra masih bagus |
| 40 dB | *Reasonable*, terdapat butiran halus seperti salju dan beberapa detail citra hilang |
| 30 dB | *Reasonable*, terdapat butiran halus seperti salju dan beberapa detail citra hilang |

PSNR digunakan untuk mengetahui perbandingan kualitas gambar sebelum dan sesudah disisipkan pesan. Untuk menentukan PSNR, terlebih dahulu harus ditentukan nilai rata-rata kuadrat dari MSE *(Mean Square Error).*

Perhitungan MSE adalah sebagai berikut :

*MSE = [I(x, y)-J(x, y)]2* (Perhitungan 1)

Keterangan:

MSE : Nilai MSE dari gambar steganografi

M : panjang gambar *stego* (dalam piksel)

N : lebar gambar *stego* (dalam piksel)

I(x,y) : nilai piksel dari gambar asli

J(x,y) : nilai piksel dari gambar *stego*

Sementara nilai PSNR dihitung dari kuadrat nilai :

PSNR=10 x log 10 (Perhitungan 2)

Keterangan :

PSNR : nilai PSNR gambar (dalam dB)

MAXi : nilai maksimum piksel gambar

MSE : nilai MSE

# 4. PEMBAHASAN

## **Hasil dan Percobaan**

20 gambar yang dijadikan sebagai *cover image* dan 15 pesan rahasia yang masing-masing kapasitas ukuranya berbeda diuji dengan memasukkan tiap gambar yang berbeda dengan tiap pesan yang berbeda secara acak. Pengujian dilakukan sebanyak 30 kali yang bertujuan untuk mengetahui seberapa besar kapasitas gambar yang akan disisipkan pesan dapat menampung pesan dan mengetahui kualitas gambar setelah disipikan pesan.

## **Pengujian *Embed***

Pengujian *embed* dilakukan untuk mengetahui daya tampung dan kualitas *image* yang akan disisipkan pesan rahasia. Berikut ini tabel *embedding* dari hasil pengujian dapat dilihat pada Tabel 3.

Tabel 3. Tabel Pengujian Embedding

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 2 | E:\SKRIPSI\image\pepper.bmp  Pepper.bmp  Dimensi:512x512  Ukuran:768KB | Ukuran:33.5KB | E:\SKRIPSI\stego_image.bmp  Stego\_pepper.bmp  Dimensi:512x512  Ukuran:768KB | 411804 | 43.2746 | Berhasil |
| 3 | E:\SKRIPSI\image\Flower.bmp  Flower.bmp  Dimensi768x512  Ukuran:1.12MB | Ukuran:100KB | - | 605297 | - | Gagal |
| 4 | E:\SKRIPSI\image\waterfall.bmp  Waterfall.bmp  Dimensi:800x600  Ukuran:1.37MB | Ukuran:100KB | E:\SKRIPSI\stego_image.bmp  Stego\_waterfall.bmp  Dimensi:800x600  Ukuran:1.37MB | 742187 | 39.6086 | Berhasil |

*Cover image* yang digunakan penelitian pada Tabel 3. masing-masing adalah sebesar 512x512 piksel berukuran 768KB untuk Airplane.bmp, 512x512 piksel berukuran 768KB untuk Pepper.bmp, 768x512 piksel berukuran 1.12MB untuk flower.bmp dan 800x600 berukuran 1.37MB untuk waterfall.bmp. Untuk pesan masing-masing berukuran 56KB, 33.5KB dan 100KB. Dari hasil pengujian no 1 dan no 2 pada Tabel 3. didapat *cover* *image* Airplane.bmp dan Pepper.bmp dengan piksel dan ukuran *file* yang sama masing-masing berhasil disisipkan pesan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | ***Cover Image*** | **Pesan Asli** | ***Stego Image*** | **Daya Tampung Pesan (bit)** | **PSNR**  ***(Peak Signal to Noise Ratio)* dB** | **Keterangan** |
| 1 | E:\SKRIPSI\image\airplane.bmp  Airplane.bmp  Dimensi:512x512  Ukuran:768KB | Ukuran:56KB | E:\SKRIPSI\stego_image.bmp  Stego\_airplane.bmp  Dimensi:512x512  Ukuran:768KB | 407922 | 41.1084 | Berhasil |

untuk *cover* Airplane.bmp sebesar 56KB dan cover Pepper.bmp sebesar 33.5KB. Didapat selisih hasil kapasitas daya tampung pesan meski piksel dan ukuran sama untuk Airplane.bmp sebesar 407922 bit dan Pepper.bmp 411084 bit.

Sedangkan untuk hasil PSNR *(Peak Signal to Noise Ratio)* dengan ukuran pesan yang berbeda ukuran didapat hasil *stego image* dari Airplane.bmp 41.1084 dB dan Pepper.bmp 43.2746 dB. Pengujian Flower.bmp 768x512 piksel berukuran 1.12MB dengan pesan berukuran 100KB gagal disisipkan dan Waterfall.bmp 800x600 piksel berukuran 1.37 berhasil disisipkan pesan berukuran 100KB. Gagalnya penyisipan pesan didalam gambar Flower.bmp karena ukuran pesan yang disisipkan terlalu besar sedangkan daya tampung gambar lebih kecil. Dengan ukuran pesan yang disisipkan sama yaitu 100KB tapi hasil kapasitas daya tampung berbeda. Untuk daya tampung pesan Waterfall.bmp dihasilkan 742187 bit, lebih besar dibandingkan dengan daya tampung Flower.bmp yaitu 605297 bit.

## **Pengujian Ekstraksi**

Selanjutnya dilakukan untuk pengujian ekstraksi pesan yaitu mengambil kembali pesan asli yang disisipkan didalam *cover image* dari pengujian embed sebelumnya. Berikut adalah tabel pengujian ekstraksi pesan dapat dilihat pada Tabel 4.

Tabel 4. Tabel Ekstraksi

*Stego image* dari Tabel 4 yang telah disisipkan pesan masing-masing stego\_jeruk.bmp 232x217 piksel berukuran 147KB, stego\_flower.bmp 768x512 piksel berukuran 1.12MB dan stego\_ikan.bmp 750x422 berukuran 928KB. Pesan yang disisipkan berukuran 1.1KB, 80.6KB dan 60.7KB. Hasil ekstraksi yang telah diuji tidak ada perubahan ukuran pesan hasil ekstraksi dengan pesan aslinya dan isi pesan dapat dibaca sesuai dengan pesan aslinya**.**

## **Perbandingan Citra Asli dengan *Stego Image***

Setelah dilakukan pengujian *embed* dan ekstraksi pesan, maka didapat hasil dari ukuran citra asli sebelum disisipkan pesan *dan stego image* setelah disisipkan pesan. Berikut ini tabel perbandingan citra asli dengan *stego image* dapat dilihat pada Tabel 5.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | ***Stego image*** | **Pesan Asli** | **Pesan Ekstraksi** | **Keterangan** |
| 1 | jeruk  Stego\_jeruk.bmp  Dimensi:232x217  Ukuran:147KB | Ukuran:1.1KB | Ukuran:1.1KB | Berhasil |
| 2 | E:\SKRIPSI\stego_image.bmp  Stego\_flower.bmp  Dimensi768x512  Ukuran:1.12MB | Ukuran:80.6KB | Ukuran:80.6KB | Berhasil |
| 3 | E:\SKRIPSI\stego_image.bmp  Stego\_ikan.bmp  Dimensi:750x422  Ukuran:928KB | Ukuran:60.7KB | Ukuran:60.7KB | Berhasil |

Tabel 5. Tabel Perbandingan Citra

|  |  |  |
| --- | --- | --- |
| **No** | ***Cover Image*** | ***Stego Image*** |
| 1 | E:\SKRIPSI\image\lighthouse.bmp  Lighthouse.bmp  Dimensi:512x768  Ukuran:1.112MB | E:\SKRIPSI\stego_image.bmp  Stego\_lighthouse.bmp  Dimensi:512x768  Ukuran:1.112MB |
| 2 | C:\Users\David Saputra\AppData\Local\Microsoft\Windows\INetCache\Content.Word\matahari.bmp  Matahari.bmp  Dimensi:284x177  Ukuran: 147KB | E:\SKRIPSI\stego_image.bmp  Stego\_matahari.bmp  Dimensi:284x177  Ukuran: 147KB |
| 3 | E:\SKRIPSI\image\Sail.bmp  Sail.bmp  Dimensi:512x768  Ukuran:1.12MB | E:\SKRIPSI\stego_image.bmp  Stego\_sail.bmp  Dimensi:512x768  Ukuran:1.12MB |

*Cover image* dari Tabel 5 sebelum disisipkan pesan masing-masing berukuran 512x768 piksel berukuran 1.12MB untuk *file* Lighthouse.bmp, 284x177 piksel berukuran 147KB untuk *file* Matahari.bmp dan Sail.bmp 512x768 piksel berukuran 1.12MB. *Hasil stego image* dari proses penyisipan pesan didalam *cover image* tidak ada perubahan ukuran, ukuran *stego image* sama dengan *cover image* sebelum disisipkan pesan.

**Perbandingan PSNR *Stego Image***

PSNR *(Peak Signal to Noise Ratio)* digunakan untuk menentukan kualitas gambar setelah pesan disisipkan. Gambar  *stego* hasil dari gambar yang sudah disisipkan pesan dibandingkan dengan gambar aslinya sebelum disisipkan pesan yang bertujuan untuk mengetahui kualitas gambar *stego*. Berikut ini tabel nilai PSNR *stego image* dapat dilihat pada Tabel 6

Tabel 6. Tabel Perbandingan PSNR

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | ***Cover Image*** | **Pesan Asli** | ***Stego Image*** | **PSNR**  ***(Peak Signal to Noise Ratio)* dB** |
| **1** | barbara  Barbaral.bmp  Dimensi:720x576  Ukuran:1.18MB | Ukuran:2.05KB | stego_barbara  Stego\_barbara.bmp  Dimensi:720x576  Ukuran:1.18MB | 54.7457 |
| **2** | E:\SKRIPSI\image\waterfall.bmp  Waterfall.bmp  Dimensi:800x600  Ukuran:1.37MB | Ukuran:100KB | E:\SKRIPSI\stego_image.bmp  Stego\_waterfall.bmp  Dimensi:800x600  Ukuran:1.37MB | 39.6086 |
| 3 | E:\SKRIPSI\image\danau.bmp  Danau.bmp  Dimensi:800x480  Ukuran:1.09MB | Ukuran:1.19KB | E:\SKRIPSI\stego_image.bmp  Stego\_danau.bmp  Dimensi:800x480  Ukuran:1.09MB | 60.625 |
| 4 | E:\SKRIPSI\image\waterfall.bmp  Waterfall.bmp  Dimensi:800x600  Ukuran:1.37MB | Ukuran:1.1KB | E:\SKRIPSI\stego_image.bmp  Stego\_waterfall1.bmp  Dimensi:800x600  Ukuran:1.37MB | 61.4191 |

Dari data Tabel 6 terdapat empat *cover image,* dua *cover* berbeda piksel dan ukuran dan dua cover dengan piksel dan ukuran yang sama. Pesan yang disisipkan terdiri dari empat pesan berbeda ukuran masing-masing yaitu 2.05KB, 100KB, 1.19KB dan 1.1KB. *Cover image* Barbara.bmp 720x576 piksel berukuran 1.18MB disisipkan pesan berukuran 2.02KB menghasilkan nilai PSNR 54.7457 dB dan *cover image* danau.bmp 800x480 piksel berukuran 1.09MB disisipkan pesan 1.19KB menghasilkan nilai PSNR 60.625 dB. Ada perbedaan kualitas nilai PSNR anatara Barbara.bmp dengan danau.bmp. Semakin besar ukuran pesan yang disisipkan maka semakin kecil hasil PSNR yang dihasilkan. Hasil piksel dan ukuran *stego image* masing-masing sama dengan ukuran *cover image* sebelum disisipkan pesan.

Dari hasil penyisipan pesan *image* yang sama dengan nama *file* Waterfall.bmp 800x600 piksel berukuran 1.37MB masing-masing disisipkan pesan dengan ukuran yang berbeda yaitu 100KB dan 1.1KB menghasilkan  *stego image* masing-masing dengan piksel dan ukuran yang sama. Hasil PSNR dari stego image masing-masing didapat hasil yang berbeda nilainya, untuk *cover image* yang disisipkan pesan berukuran 100KB sebesar 39.6086 dB dan pesan berukuran 1.1KB sebesar 61.4191 dB. Nilai PSNR *stego image* Waterfall1.bmp megalami peningkatan terhadap *stego image* Waterfall.bmp setelah disisipkan pesan yang berukuran 1.1KB yang sebelumnya disisipkan pesan 100KB.

# 5. KESIMPULAN

Dari penelitian yang dilakukan, kesimpulan yang dapat diambil sebagai berikut:

1. *Cover image* ukuran 800x600 dapat menampung pesan 742.187 bit dan *cover image* ukuran 284x177 dapat menampung pesan 73.318 bit. Semakin besar nilai selisih piksel maka semakin besar kapasitas daya tampung pesan sebaliknya jika selisih kedua piksel *cover* image kecil maka semakin kecil kapasitas daya tampung pesan *cover image* untuk disisipkan pesan.

2. *Cover image* 800x600 disisipkan pesan masing-masing berukuran 1.1KB dan 100KB menghasilkan nilai PSNR 61,4191 dan 39,6086. Nilai PSNR *stego image* tergantung dari besarnya pesan yang disisipkan dalam gambar, semakin besar nilai PSNR maka kualitas *stego image* semakin baik dan apabila nilai PSNR semakin kecil maka kualitas *stego image* semakin buruk karena mengalami penurunan hasil citra hal ini juga berpengaruh terhadap besarnya ukuran pesan yang disisipkan.

3. Dari 30 data yang diuji, keberhasilan sistem dalam menyisipkan pesan mencapai 100% dengan rincian 27 cover image berhasil disisipkan pesan dan 3 *cover image* gagal disisipkan pesan rahasia, kegagalan proses penyisipan karena ukuran pesan lebih besar di bandingkan kapasitas daya tampung *cover image*

# 6. UCAPAN TERIMA KASIH

Terimakasih kepada seluruh pihak yng ikut terlibat diantaranya dosen pembimbing, dosen FTI Universitas Mercu Buana Yogyakarta serta teman-teman FTI 2015

**DAFTAR PUSTAKA**

Achmad, S., Hakim, Z. & Permana, E. A., 2014. Analisis Dan Implementasi Teknik Steganografi Sebagai Fasilitas Pengamanan Proses Pengiriman File Secara Online. Sisfotek Global, Vol.4 No.1/Maret 2014 ISSN: 2088 – 1762.

Andono, P. N., T. & M., 2017. Pengolahan Citra Digital. pp. 82-86, ISBN: 978-979-29-6370-0.

Azhary, M., 2014. Pengembangan Aplikasi Steganografi Pixel Value Differencing (PVD). Seminar Nasional Sains dan Teknologi 2014, pp. ISSN: 2407 - 1846.

Haryanto, E., 2016. Implementasi Teknik Steganografi Sebagai Anti Forensik Penyisipan Teks Pada Citra. Jurnal Informasi Interaktif, pp. ISSN: 2527-5240.

Martsanto, S. & Jazuli, W., 2016. Teknik Steganografi Dan Enkripsi Dokumen Guna Menjamin Keamanan Dan Integritas Informasi Dalam Lingkup Organisasi (Studi Kasus Pada Pt Saptawara Teknologi Indonesia). Jurnal Ilmiah Ilmu Komputer, pp. ISSN: 2503-3832.

Muhammad, R., Setyaningsih, F. A. & Diponegoro, M., 2018. Analisis Kompresi Steganography Pada Citra Digital Dengan Menggunakan Metode Least Significant Bit Berbasis Mobile. Jurnal Coding, Rekayasa Sistem Komputer, pp. ISSN: 2338-493X.

Niswati, Z., 2014. Steganografi Berbasis Least Significant Bit (LSB) Untuk Menyisipkan Gambar Ke Dalam Citra Gambar. Faktor Exacta Vol. 5, pp. 181-191, ISSN: 1979 276X.

Nizirwan, A., 2018. Perancangan Steganografi Hidden Message Dengan Metode Least Significant Bit Insertion (LSB) Berbasis. Jurnal Algoritma, Logika dan Komputasi, pp. ISSN: 2620-620X.

Rahim, R., 2016. Penyisipan Pesan Dengan Algoritma Pixel Value Differences Dengan Algoritma Caesar Cipher Pada Proses Steganografi. Jurnal TIMES , Vol. V No 1, pp. 6-11, ISSN: 2337 - 3601.

Sinaga, H. A. Y. & Sitorus, L., 2017. Pengamanan File Citra Digital Dengan Menggunakan Metode Least Significant Bit Dan End Of File. Jurnal Teknik Informatika Unika St. Thomas (JTIUST, pp. , Volume 02 Nomor 02, ISSN: 2548-1916.

Siringoringo, R., 2016. Analisis Psnr Pada Steganografi Least Significant Bit Dengan Pesan Terenkripsi Advanced Encription System. Jurnal Methodika, Vol. 2 No. 1 ISSN: 2442-7861.

Za'iwatun, N., 2014. Steganografi Berbasis Least Significant Bit (LSB) Untuk Menyisipkan Gambar Ke Dalam Citra Gambar. Faktor Exacta Vol. 5 No. 2, pp. ISSN: 1979 276X.