Volume 2, No.2, Nopember 2016

p-ISSN : 2443-2245
e-ISSN : 2443-2334

MULTINETICS

JURNAL MULTIMEDIA NETWORKING INFORMATICS

JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER

TABLE OF CONTENTS

ARTICLES

EDITORS

Perancangan Aplikasi Steganografi Berbasis Android dengan Metode *Pixel Value Differencing* (PVD)

Estu Sinduningrum, Anton Supriyanto
Fakultas Teknik Program Studi Informatika
Universitas Muhammadiyah Prof. Dr. Hamka
Jl. Tanah Merdeka no.6, Kp. Rambutan, Ps. Rebo, Jakarta Timur
etu_s2@yahoo.com, anton_supriyanto@outlook.co.id

Diterima: 13 September 2016. Disetujui: 27 Oktober 2016. Dipublikasikan: Nopember 2016

***Abstract* - Simplicity in message exchanges on current technological developments. But on the other hand it brings new problems in terms of security of the message. One method used to overcome this problem is steganography. Steganography is a technique for hiding information on an image. The purpose of this study is to analyze the quality of the images used steganography to hide text messages. One of steganography techniques to be used in this research is Pixel Value Differencing (PVD). Quality of the image of will be declined. The application will be tested using .bmp, .png and .jpg images types and quality of steganography image is tested by calculating the Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR) results of the insertion process of the message. The results of the study showed that the characteristics of the image greatly affect the capacity of the message and the quality of the image after embedded message. Images with different sizes and types tested using MSE get values between 1.62022 dB and 0.000116 dB and using PSNR get values between 46.0 dB and 90.0 dB. This shows that the qualities between the original images and the steganography images are not much different. It can be concluded that the quality of the image are slightly different.**

Keywords: steganography, pixel value differencing, mse, psnr

I. PENDAHULUAN

Teknologi dimaksudkan untuk memudahkan segala bentuk kegiatan kita sehari-hari. Saat ini perkembangan teknologi yang semakin meningkat dari *handphone* memberikan dampak besar terhadap penggunaannya[1].

Salah satu sistem operasi yang digunakan oleh *smartphone* adalah Android. Kelebihan Android dibanding sistem operasi *smartphone* lainnya karena Android bersifat *open source* sehingga memudahkan para pengembang untuk menciptakan dan memodifikasi aplikasi atau fitur-fitur yang belum ada di sistem Android sesuai dengan keinginan mereka sendiri.

Berbagai macam teknik digunakan untuk melindungi pesan yang dirahasiakan dari orang yang tidak berhak dengan salah satu teknik ini yaitu *Steganography*, teknik penyembunyian pesan pada media digital[2].

Steganografi dapat menyembunyikan pesan di berbagai media digital, yaitu gambar, suara dan video atau dalam format lainnya. Beberapa metode telah dikembangkan untuk steganografi diantaranya *Least Significant Bits (LSB), Pixel Value Differencing (PVD), Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT)* [3]. Pada penelitian *Least Significant Bits (LSB)* metode ini banyak dilakukan dengan pendekatan yang sederhana menyisipkan suatu informasi pada suatu media dengan mengganti nilai-nilai bit dengan bit data yang ingin disisipkan. Namun teknik ini memiliki kelemahan jika sebuah pesan yang akan disisipkan jumlahnya besar maka hasil dari media yang telah disisipkan akan mengalami distorsi besar.

Salah satu metode algoritma yang akan digunakan pada penelitian ini adalah *Pixel Value Differencing* (*PVD*) dikarenakan metode ini dikembangkan untuk meningkatkan daya tampung pesan dan pengurangan tingkat distorsi pada steganografi[4]. Cara kerja teknik *Pixel Value Differencing* (*PVD*) adalah dengan cara membagi media yang akan disisipkan menjadi blok-blok piksel yang bertetangga[5]. Blok-blok tersebut terdiri dari dua buah *pixel* yang posisinya berdekatan[6]. Bit–bit pesan yang akan disisipkan dihitung dengan besarnya kedua piksel tersebut.

Berdasarkan kelebihan sistem operasi Android dan metode *Pixel Value Differencing (PVD)*, maka dalam penelitian ini akan dibuat Perancangan Aplikasi Steganografi Berbasis Android Dengan Metode *Pixel Value Differencing (PVD)* untuk proses penyisipan pesan dan ekstraksi pesan[7].

II. LANDASAN TEORI

A. Citra Digital

Citra *(image)* adalah gambar pada bidang dua dimensi. Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dua dimensi. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut[8].

Citra (*image*) adalah kombinasi antara titik, garis, bidang dan warna untuk menciptakan suatu imitasi dari suatu objek, biasanya objek fisik atau manusia. Citra bisa berwujud gambar (*picture*) dua dimensi seperti lukisan, foto, dan yang berwujud tiga dimensi seperti, patung[2]. Adapun format *file* gambar adalah :

a) Bitmap (BMP)
   1) Merupakan format citra yang baku dilingkungan sistem Microsoft Windows dan IBM OS/2.
   2) Kualitas BMP lebih baik dan dengan ukuran yang lebih baik dari format JPG/ JPEG dan GIF.
   3) Format *file* Bitmap versi baru dari Microsoft Windows, setiap berkas/*file* terdiri dari: *header file, header bitmap*, informasi palet, dan data bitmap.

b) *Joint Photographic Experts Group (JPEG)*
   1) Dikembangkan oleh para ahli fotografi untuk mendapatkan gambar yang berukuran rasional tapi tetap menyimpan persepsi gambar yang baik.
   2) Bersifat *lossy* dengan tingkat *lossness* yang dapat diatur.
   3) Bagus untuk mengkompresi foto-foto natural tetapi kurang cocok untuk *computer-generated images* (CGI).

c) Portable Graphics Network (PNG)
   1) PNG digunakan di internet dan merupakan format "pengganti" GIF, setelah GIF terkena patent LZW yang dilakukan oleh Unisys.
   2) Diprakarsai oleh Thomas Boutell dari PNG Development Group, dan versi finalnya di-*release* pada 1 Oktober 1996.
   3) Memiliki kedalaman warna 48-bit.

B. Steganografi

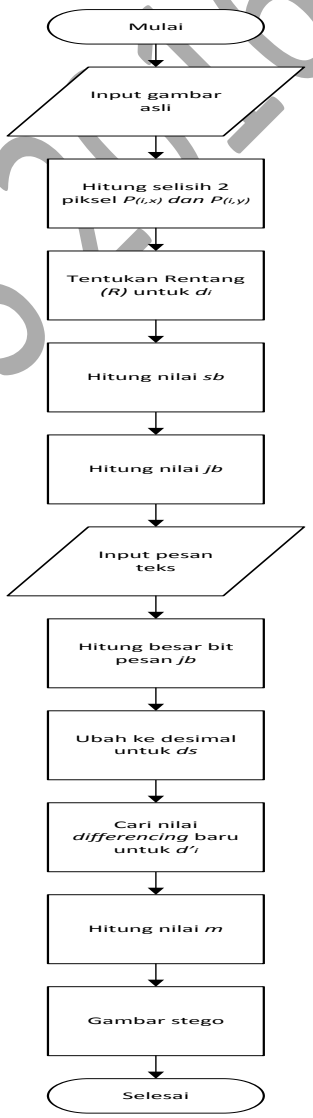
Steganografi *(steganography)* adalah ilmu dan seni menyembunyikan pesan rahasia *(hiding message)* sedemikian sehingga keberadaan pesan tidak terdeteksi oleh indera manusia[9]. Steganorafi berasal dari Bahasa Yunani yang berarti "tulisan tersembunyi" *(covered writing)*. Steganografi membutuhkan wadah penampung dan data rahasia yang akan disembunyikan [10].

1. Algoritma *Pixel Value Differencing (PVD)*

Berikut adalah aturan-aturan persamaan algoritma *Pixel Value Differencing* (PVD):

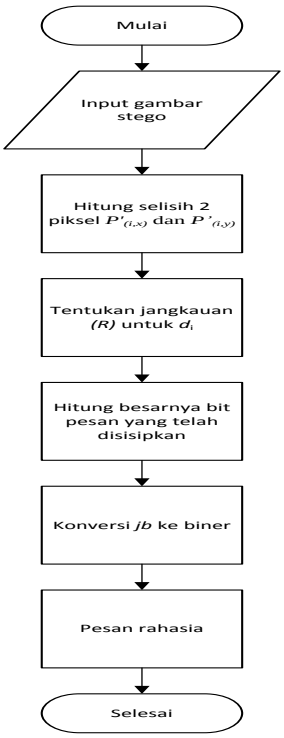
$$(P'_{(i,x)},(P'_{(i,y)}) = \begin{cases} (P_{(i,x)} + [m/2], P_{(i,y)} - [m/2]), \\ \text{Jika } P_{(i,x)} \geq P_{(i,y)} \text{ dan } d'_i > d_i; \\ (P_{(i,x)} - [m/2], P_{(i,y)} + [m/2]), \\ \text{Jika } P_{(i,x)} < P_{(i,y)} \text{ dan } d'_i > d_i; \\ (P_{(i,x)} - [m/2], P_{(i,y)} + [m/2]), \\ \text{Jika } P_{(i,x)} \geq P_{(i,y)} \text{ dan } d'_i \leq d_i; \\ (P_{(i,x)} + [m/2], P_{(i,y)} - [m/2]), \\ \text{Jika } P_{(i,x)} < P_{(i,y)} \text{ dan } d'_i \leq d_i; \end{cases}$$

Algoritma Penyisipan Pesan



Gambar 1. *Flowchart* Penyisipan

Algoritma Ekstraksi Pesan



Gambar 2. *Fowchart* Ekstraksi

2. *Peak Signal to Noise Ratio (PNSR)* dan *Mean Square Error (MSE)*

PSNR 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 1 memperlihatkan nilai PSNR beserta penjelasannya.

TABEL 1. 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	*Poor*, terdapat banyak derau pada citra
20 dB	*Unusable*

Berikut adalah sebuah tabel jangkauan yang digunakan untuk menentukan banyaknya bit yang akan disisipkan:

TABEL 2 BIT YANG DAPAT DISISIPKAN PADA DAERAH RENTANG

R (Rentang)	1	2	3	4	5	6
bb (batas bawah)	0	8	16	32	64	128
ba (batas atas)	7	15	31	63	127	255
jb (jumlah bit)	3	3	4	5	6	7

Pada penelitian ini, 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:

$$\text{MSE} = \frac{1}{MN} \sum_{y-1}^{M} \sum_{x-1}^{N} [I(x,y) - J(x,y)]^2 \qquad (1)$$

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 maksimum sinyal dibagi dengan MSE.

$$\text{PSNR} = 10\ x\ log_{\ 10} \left(\frac{MAXi}{MSE}\right) \qquad (2)$$

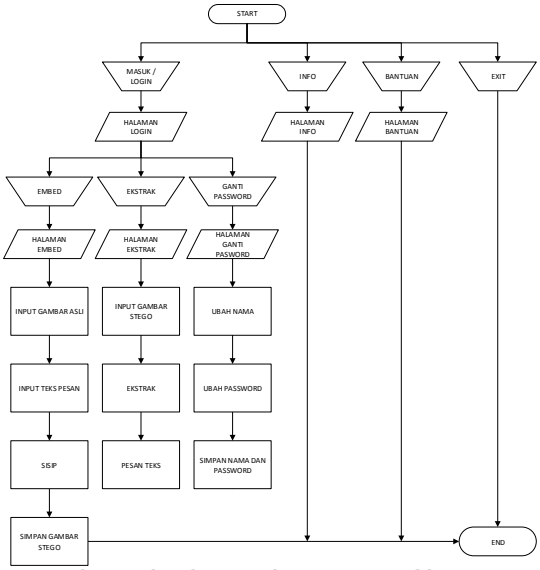
Keterangan:

PSNR : nilai PSNR gambar (dalam dB)
MAXi : nilai maksimum piksel gambar
MSE : nilai MSE

III. PERANCANGAN SISTEM

Metode yang digunakan dalam perancangan sistem yaitu *flowchart* dan *storyboard* yang menggambarkan tentang proses yang terjadi pada sistem yaitu proses masuk *(login)*, penyisipan pesan *(embedding)*, ubah *password (update password)*, dan ekstrak pesan *(ekstract)*.

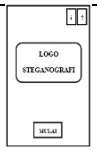
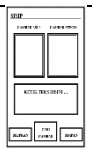
A. Flowchart Gambaran Umum Aplikasi

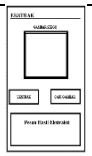


Gambar 3. *Flowchart* Gambaran Umum Aplikasi

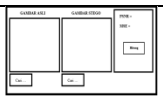
B. Storyboard Aplikasi

TABEL 3 STORYBOARD APLIKASI ANDROID

No	Nama	Desain	Keterangan
1	*Splash Screen*		• Halaman ini menampilkan **Logo** aplikasi sebagai branding aplikasi. • Durasi 3-5 detik.
2	Halaman Awal		• Pada halaman awal terdapat **Logo** aplikasi di tengah, tombol **Mulai** di bawah, tombol **(i)** dan tombol **(?)** dipojok kanan atas. • Tombol **Mulai** untuk menuju ke halaman **Login** yang digunakan untuk menginput nama dan password pengguna. • Tombol **(i)** untuk menuju ke halaman **Info** yang menampilkan informasi/versi dari aplikasi. • Tombol **(?)** untuk menuju ke halaman **Bantuan** yang menampilkan isi petunjuk penggunaan aplikasi.
3	Halaman Info		• Pada pojok kiri atas terdapat judul halaman. • Halaman ini menampilkan versi dari aplikasi.
4	Halaman Bantuan		• Pada pojok kiri atas terdapat judul halaman. • Halaman ini berisi tentang petunjuk penggunaan aplikasi.
5	Halaman Login		• Pada pojok kiri atas terdapat judul halaman. • Dibagian tengah terdapat **Nama** dan **Password**. • Dibagian bawah terdapat tombol **Masuk**.
			• **Nama** digunakan untuk menginput nama pengguna. • **Password** digunakan untuk menginput password pengguna. • Tombol **Masuk** untuk memproses akun pengguna menuju ke halaman **Menu**.
6	Halaman Menu		• Pada pojok kiri atas terdapat judul halaman. • Dibagian tengah terdapat **Logo** aplikasi. • Dibagian bawah terdapat tombol **Ekstrak**, tombol **Ganti Password** dan tombol **Sisip**. • Tombol **Ekstrak** untuk menuju ke halaman **Ekstrak.** • Tombol **Ganti Password** untuk menuju ke halaman **Ganti Password**. • Tombol **Sisip** untuk ke halaman **Sisip**.
7	Halaman Ganti Password		• Pada pojok kiri atas terdapat judul halaman. • Dibagian tengah terdapat **Nama lama, Nama baru, Password lama, Password baru** dan tombol **Ganti**. • **Nama lama** merupakan nama dari pengguna awal. • **Nama baru** digunakan untuk nama pengguna baru. • **Password lama** merupakan password dari pengguna awal. • **Password baru** digunakan untuk password pengguna baru. • Tombol **Ganti** untuk memproses/ mengupdate pengguna baru.
8	Halaman Sisip		• Pada pojok kiri atas terdapat judul halaman. • Dibagian tengah terdapat kotak **Gambar asli,** kotak **Gambar stego** dan kotak **Pesan.** • Dibagian bawah terdapat tombol **Sisip,** tombol **Cari Gambar** dan tombol **Simpan.** • Kotak **Gambar asli** menampilkan gambar asli yang akan digunakan untuk menyisipkan pesan. • Kotak **Gambar stego** menampilkan gambar stego hasil dari penyisipan pesan. • Kotak **Pesan** digunakan untuk mengisi pesan teks yang akan disisipkan ke gambar asli. • Tombol **Sisipkan** digunakan untuk memproses penyisipan pesan ke dalam gambar asli. • Tombol **Cari Gambar** digunakan untuk mencari gambar asli dari

			penyimpanan internal. • Tombol **Simpan** digunakan untuk menyimpan gambar stego ke penyimpanan internal.
9	Halaman Ekstrak		• Pada pojok kiri atas terdapat judul halaman. • Dibagian tengah terdapat kotak **Gambar Stego**, tombol **Ekstrak** dan tombol **Cari Gambar**. • Dibagian bawah terdapat kotak **Pesan** dari hasil ekstraksi. • Tombol **Cari Gambar** digunakan untuk mencari gambar stego yang akan di ekstraksi dalam penyimpanan internal. • Kotak **Gambar Stego** akan menampilkan gambar yang telah disisipkan pesan. • Tombol **Ekstrak** digunakan untuk proses ekstraksi pesan dari gambar stego. • Kotak **Pesan** akan menampilkan isi pesan hasil ekstraksi dari gambar stego.

TABEL 4. STORYBOARD APLIKASI MATLAB

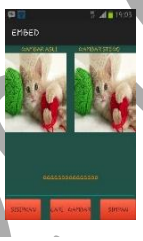
No	Nama	Desain	Keterangan
1	Halaman Menu		• Halaman menu terdapat kotak **Gambar Asli**, **Gambar Stego**, **Psnr Mse**, dan tombol **Cari**. • Kotak **Gambar Asli** akan menampilkan gambar sebelum pesan disisipkan. • Kotak **Gambar Stego** akan menampilkan gambar yang telah disisipkan pesan. • Tombol **Cari** digunakan untuk mencari gambar asli atau gambar stego yang akan di hitung nilai psnr dan msenya. • Tombol **Hitung** digunakan untuk proses perhitungan nilai psnr dan mse dari gambar asli dan gambar stego dan kemudian hasilnya ditampilkan.

IV. IMPLEMENTASI DAN PENGUJIAN

Pengujian sistem ini dilakukan dengan pendekatan *Desain test case*, yaitu dengan menguji aplikasi dengan mencari fungsi–fungsi yang salah, kesalahan desain *interface*, kesalahan struktur berkas atau akses eksternal. *Black-box* berfokus pada persyarakatan fungsional perangkat lunak, sehingga mendapatkan serangkaian kondisi input yang seluruhnya menggunakan syarat fungsional pada suatu program *(MSE PSNR)*.

A. *Pengujian Aplikasi*

Gambar 4. Halaman *Splash Screen*



Gambar 5. Proses Penyisipan

Gambar 6. Proses Ekstraksi

B. *Pengujian Gambar*

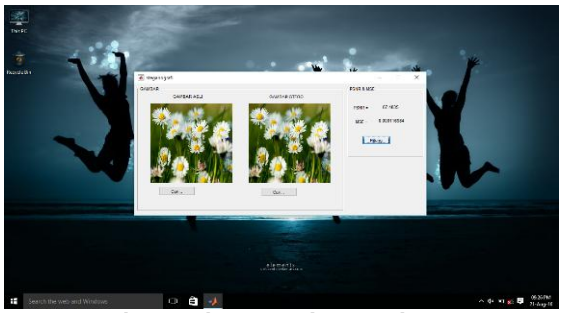
Pada penelitian ini terdapat enam buah gambar dengan beberapa tipe dan ukuran yang digunakan

TABEL 5. GAMBAR YANG AKAN DI UJI DENGAN UKURAN PIKSEL BERVARIASI

Nama Gambar	Gambar	Ukuran Gambar (piksel)	Tipe Gambar	Ukuran gambar (KB)
Bunga		128 x 128	PNG	42.7
Ferrari		128 x 128	JPG	10.7
Singa		128 x 128	BMP	48.0

C. *Pengujian Nilai MSE dan PSNR*

Algoritma MSE dan PSNR dibuat dengan menggunakan bahasa pemrograman Matlab. Matlab merupakan bahasa pemrograman yang sangat baik untuk mengolah gambar karena dilengkapi fungsi-fungsi yang memudahkan pemakaiannya. Berikut ini merupakan program yang digunakan untuk mengetahui nilai MSE dan PSNR dari setiap gambar yang sebelum dan sesudah disisipkan pesan



Gambar 7. Perhitungan Nilai PSNR dan MSE

TABEL 6. GAMBAR YANG AKAN DI UJI DENGAN UKURAN PIKSEL SAMA

Nama Gambar	Gambar	Ukuran Gambar (piksel)	Tipe Gambar	Ukuran gambar (KB)
Bunga		512 x 512	PNG	734
Kucing		912 x 800	BMP	2000
Singa		128 x 116	JPG	10.9

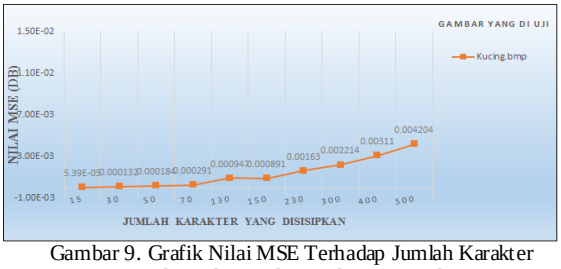
D. *Hasil Pengujian*

Hasil pengujian nilai MSE pada gambar dengan ukuran piksel dan tipe yang berbeda dapat dilihat pada Tabel 7.

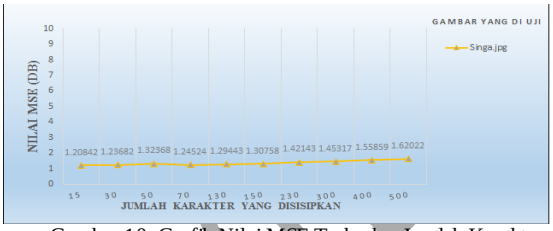
TABEL 7. HASIL PENGUJIAN GAMBAR DALAM NILAI MSE (DB)

Jumlah karakter yang disisipkan	NILAI MSE (db)		
	Bunga.png	Kucing.bmp	Singa.jpg
15	0.000116	0.000053	1.20842
30	0.000389	0.000132	1.23682
50	0.000729	0.000184	1.32368
70	0.001118	0.000291	1.24524
130	0.002301	0.000947	1.29443
150	0.003281	0.000891	1.30758
230	0.003607	0.00163	1.42143
300	0.007386	0.002214	1.45317
400	0.010397	0.00311	1.55859
500	0.013989	0.004204	1.62022

Gambar 8. Grafik Nilai MSE Terhadap Jumlah Karakter yang disisipkan pada Gambar Bunga.png



Gambar 9. Grafik Nilai MSE Terhadap Jumlah Karakter yang disisipkan pada Gambar Kucing.bmp

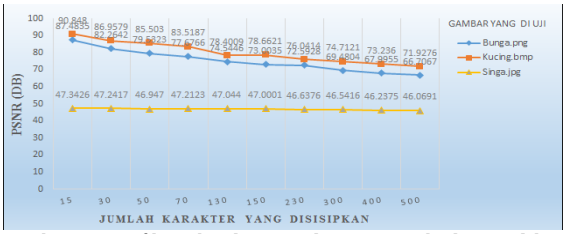


Gambar 10. Grafik Nilai MSE Terhadap Jumlah Karakter yang disisipkan pada Gambar Singa.jpg

Hasil pengujian nilai PSNR pada gambar dengan ukuran piksel dan tipe yang berbeda dapat dilihat pada Tabel 8.

TABEL 8. HASIL PENGUJIAN GAMBAR DALAM NILAI PSNR (DB)

Jumlah karakter yang disisipkan	NILAI PSNR (db)		
	Bunga.png	Kucing.bmp	Singa.jpg
15	87.4835	90.848	47.3426
30	82.2642	86.9579	47.2417
50	79.5323	85.503	46.947
70	77.6766	83.5187	47.2123
130	74.5446	78.4009	47.044
150	73.0035	78.6621	47.0001
230	72.5928	76.0414	46.6376
300	69.4804	74.7121	46.5416
400	67.9955	73.236	46.2375
500	66.7067	71.9276	46.0691



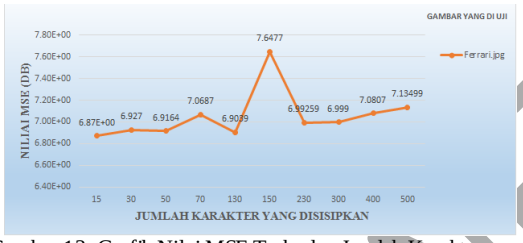
Gambar 11. Grafik Perbandingan Nilai PSNR Terhadap Jumlah Karakter yang disisipkan pada Gambar Bunga.png, Kucing.bmp dan Singa.jpg

Hasil pengujian nilai MSE pada ketiga gambar dengan ukuran piksel yang sama dapat dilihat pada Tabel 9.

TABEL 9. HASIL PENGUJIAN GAMBAR DALAM NILAI MSE (DB)

Jumlah karakter yang disisipkan	NILAI MSE (db)		
	Bunga.png	Ferrari.jpg	Singa.bmp
15	0.003031	6.8701	0.00921
30	0.006327	6.927	0.01747
50	0.013590	6.9164	0.03108
70	0.379171	7.0687	0.04317
130	0.39679	6.9039	0.08593
150	0.060852	7.6477	0.10424
230	0.069925	6.99259	0.16278
300	0.128906	6.999	0.22363
400	0.139201	7.0807	0.28945
500	0.223897	7.13499	0.52794

Gambar 12. Grafik Nilai MSE Terhadap Jumlah Karakter yang disisipkan pada Gambar Bunga.png



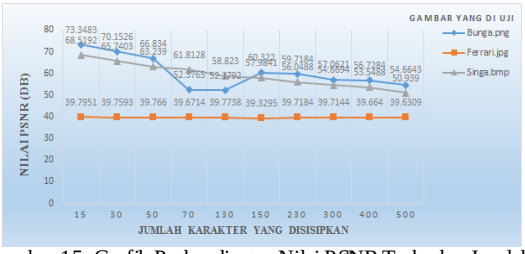
Gambar 13. Grafik Nilai MSE Terhadap Jumlah Karakter yang disisipkan pada Gambar Ferrari.jpg

Gambar 14. Grafik Nilai MSE Terhadap Jumlah Karakter yang disisipkan pada Gambar Singa.bmp

Hasil pengujian nilai PSNR pada ketiga gambar dengan ukuran piksel yang sama:

TABEL 10. HASIL PENGUJIAN GAMBAR DALAM NILAI PSNR (DB)

Jumlah karakter yang disisipkan	NILAI PSNR (db)		
	Bunga.png	Ferrari.jpg	Singa.bmp
15	73.3483	39.7951	68.5192
30	70.1526	39.7593	65.7403
50	66.834	39.766	63.239
70	52.3765	39.6714	61.8128
130	52.1792	39.7738	58.823
150	60.322	39.3295	57.9841
230	59.7184	39.7184	56.0488
300	57.0621	39.7144	54.6694
400	56.7284	39.664	53.5488
500	54.6643	39.6309	50.939



Gambar 15. Grafik Perbandingan Nilai PSNR Terhadap Jumlah Karakter yang disisipkan pada Gambar Bunga.png, Ferrari.jpg dan Singa.bmp

E. *Pengujian Stress Testing*

Stress *testing* adalah salah satu jenis pengujian sistem *(system testing)*. Pengujian ini bertujuan untuk melihat apakah perangkat lunak secara keseluruhan mampu menangani kebutuhan sumberdaya yang tidak normal (mencakup kuantitas, frekuensi, maupun *volume*). Apakah data dalam jumlah sangat besar, dengan frekuensi sangat tinggi, serta *volume* yang sangat besar mengakibatkan performa atau bahkan fungsionalitas perangkat lunak terganggu atau tidak. Jadi, meskipun perangkat lunak anda sudah berjalan baik di setiap fiturnya, pastikan juga setiap perangkat lunak tetap bekerja dengan maksimal saat diberi 'beban berat'. Jangan hanya menguji dengan data uji yang terbatas[1].

Stress test dilakukan secara bertahap yaitu pertama dengan *test* kecil dan dilanjutkan dengan maksimal yang bisa ditangani oleh aplikasi, ini penting karena kita harus mengetahui seberapa besar aplikasi dalam menangani beban.

TABEL 11 HASIL PENGUJIAN STRESS TESTING

Nama Gambar	Gambar	Ukuran (piksel)	Tipe	Ukuran (KB)	Maks pesan (karakter)
Bunga		512 x 512	PNG	734	526484
Kucing		912 x 800	BMP	2080	1576067
Singa		128 x 116	JPG	10.9	8093
Bunga		128 x 128	PNG	42.7	32178
Ferrari		128 x 128	JPG	10.7	8992

Singa		128 x 128	BMP	48.0 KB	34244

V. KESIMPULAN DAN SARAN

A. Kesimpulan

1. Aplikasi steganografi dengan metode *Pixel Value Differencing (PVD)* pada *mobile phone* berbasis Android ini dapat menyembunyikan informasi dengan melakukan penyisipan pesan dan ekstraksi pesan rahasia.
2. Teknik steganografi dengan metode *Pixel Value Differencing (PVD)* dapat digunakan pada sebuah gambar dengan 3 tipe berbeda yaitu: .png, .jpg dan .bmp dengan ukuran yang berbeda.
3. Berdasarkan dari hasil analisa penelitian ini dapat disimpulkan bahwa:
   a. Aplikasi steganografi yang dibuat dari implementasi algoritma *Pixel Value Differencing (PVD)* menunjukkan bahwa kualitas terhadap ketiga gambar yang diujikan dengan ukuran dan tipe berbeda masih dalam keadaan baik yaitu berkualitas *reasonable* sampai dengan *excelent,* karena PSNR berkisar antara 46.0 db sampai 90.8 db yaitu menunjukan perubahan kualitas gambar rendah.
   b. Pada proses ekstraksi, pesan yang disisisipkan pada gambar dalam aplikasi steganografi ini, dapat diekstrak kembali dengan baik yaitu pesan yang disisipkan sebelum proses penyisipan dan setelah proses ekstraksi sama tanpa ada perubahan atau *error* yang menyebabkan isi pesan tidak dapat dikembalikan.
   c. Hasil pengujian nilai MSE dan PSNR terhadap gambar yang dihasilkan dari aplikasi steganografi ini pun menunjukkan nilai yang cukup baik bergantung pada tipe gambar yang digunakan dan besarnya jumlah karakter yang disisipkan pada gambar tersebut. Hasil pengujian pada gambar dengan ukuran berbeda, gambar Kucing.bmp memperoleh nilai desibel yang lebih baik antara 71.0 sampai 90.0 dibandingkan dengan gambar Bunga.png antara 66.0 sampai 87.0 dan gambar Singa.jpg antara 46.0 sampai 47.0. Ini menunjukkan bahwa diantara gambar Bunga.png dan gambar Singa.jpg, gambar Kucing.bmp lebih baik untuk penyisipan pesan rahasia dengan perubahan kualitas gambar sangat kecil karena memperoleh nilai PSNR yang tinggi, semakin besar nilai PSNR semakin kecil perubahan kualitas gambar.

B. Saran

Aplikasi ini bisa dikembangkan lagi untuk penelitian selanjutnya terutama untuk mengirimkan dan menerima pesan gambar hasil steganografi.

1. Untuk lebih mengamankan data/informasi yang sangat rahasia dapat menggunakan penggabungan antara teknik steganografi dan kriptografi.
2. Pengembangan selanjutnya dapat menggunakan algoritma steganografi lain yang dapat mengkompresi sebuah gambar dengan *input* dan *output* gambar yang mempunyai format lebih bervarian lagi.
3. Aplikasi ini hanya dapat menyisipkan sebuah pesan, sehingga untuk penelitian selanjutnya dapat menggunakan *file* dengan format/tipe lainnya *.doc, *.ppt, *.pdf dan sebagainya

REFERENSI

[1] Jayadi, "Kompasiana," 5 Januari 2011. [Online]. Available:http://www.kompasiana.com/highspeed/stress-test. [Accessed 1 Agustus 2016].

[2] R. Munir, "Steganografi dan Watermarking," Bahan Kuliah ke-7, 2004.

[3] Mehdi Hussain; Ainuddin Wahid Abdul Wahab; Nor Badrul Anuar; Rosli Salleh; Rafidah Md Noor, 2015, "Pixel value differencing steganography techniques: Analysis and open challenge", IEEE International Conference on Consumer Electronics - Taiwan, Pages: 21 – 22.

[4] Zhenhao Zhu; Tao Zhang; Pengwei Zhu; Baoji Wan; Xiaodan Hou, 2013, "Steganalysis of AE-LSB steganography based on pixel value differencing", Ninth International Conference on Natural Computation (ICNC), Pages: 1449 – 1453.

[5] C. Wu and W. H. Tsai, "A steganographic method for images by pixel-value differencing", vol. 24, pp. 1613-1626, 2003.

[6] K. Gulve Avinash; M. S. Joshi, 2012, "A Secured Five Pixel Pair Differencing Algorithm for Compressed Image Steganography", Third International Conference on Computer and Communication Technology, Pages: 278 – 282.

[7] Rozali, S. Guritman and H. T. Natalisa, "Perbaikan Dan Evaluasi Kinerja Algoritma Pixel- Value Differencing (PVD)", vol. 09, 2009.

[8] R. Munir, "Pengantar Pengolahan Citra". Bandung, 2004.

[9] W. T. Hsien and S. L. Hui, "A Steganographic Method Based on Pixel-Value Differencing and the Perfect Square Number," Journal of Applied Mathematics, vol. 2013, no. 5, pp. 8, 2013.

[10] M. A. Andriawan , S. and S. J. I. Ismail, "Implementasi Steganografi Pada Citra Digital File Gambar Bitmap (BMP) Menggunakan Java", Bandung: Politeknik Telkom.

Estu Sinduningrum - Perancangan Aplikasi Steganografi Berbasis Android dengan Metode Pixel Value Differencing (PVD)

by Estu Sinduningrum Uploaded By Lutfan Zulwaqar

Submission date: 24-Feb-2020 11:21AM (UTC+0700)
Submission ID: 1262813646
File name: 1.Gabungan_Perancangan_Aplikasi_Steganografi_Berbasis_2016.pdf (1.33M)
Word count: 3696
Character count: 20766

Volume 2, No.2, Nopember 2016

p-ISSN : 2443-2245
e-ISSN : 2443-2334

MULTINETICS

JURNAL MULTIMEDIA NETWORKING INFORMATICS

JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER

Perancangan Aplikasi Steganografi Berbasis Android dengan Metode *Pixel Value Differencing* (PVD)

Estu Sinduningrum, Anton Supriyanto
Fakultas Teknik Program Studi Informatika
Universitas Muhammadiyah Prof. Dr. Hamka
Jl. Tanah Merdeka no.6, Kp. Rambutan, Ps. Rebo, Jakarta Timur
etu_s2@yahoo.com, anton_supriyanto@outlook.co.id

Diterima: 13 September 2016. Disetujui: 27 Oktober 2016. Dipublikasikan: Nopember 2016

***Abstract* - Simplicity in message exchanges on current technological developments. But on the other hand it brings new problems in terms of security of the message. One method used to overcome this problem is steganography. Steganography is a technique for hiding information on an image. The purpose of this study is to analyze the quality of the images used steganography to hide text messages. One of steganography techniques to be used in this research is Pixel Value Differencing (PVD). Quality of the image of will be dedined. The application will be tested using .bmp, .png and .jpg images types and quality of steganography image is tested by calculating the Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR) results of the insertion process of the message. The results of the study showed that the characteristics of the image greatly affect the capacity of the message and the quality of the image after embedded message. Images with different sizes and types tested using MSE get values between 1.62022 dB and 0.000116 dB and using PSNR get values between 46.0 dB and 90.0 dB. This shows that the qualities between the original images and the steganography images are not much different. It can be concluded that the quality of the image are slightly different.**

Keywords: steganography, pixel value differencing, mse, psnr

I. PENDAHULUAN

Teknologi dimaksudkan untuk memudahkan segala bentuk kegiatan kita sehari-hari. Saat ini perkembangan teknologi yang semakin meningkat dari *handphone* memberikan dampak besar terhadap penggunaannya[1].

Salah satu sistem operasi yang digunakan oleh *smartphone* adalah Android. Kelebihan Android dibanding sistem operasi *smartphone* lainnya karena Android bersifat *open source* sehingga memudahkan para pengembang untuk menciptakan dan memodifikasi aplikasi atau fitur-fitur yang belum ada di sistem Android sesuai dengan keinginan mereka sendiri.

Berbagai macam teknik digunakan untuk melindungi pesan yang dirahasiakan dari orang yang tidak berhak dengan salah satu teknik ini yaitu *Steganography*, teknik penyembunyian pesan pada media digital[2].

Steganografi dapat menyembunyikan pesan di berbagai media digital, yaitu gambar, suara dan video atau dalam format lainnya. Beberapa metode telah dikembangkan untuk steganografi diantaranya *Least Significant Bits (LSB)*, *Pixel Value Differencing (PVD)*, *Discrete Cosine Transform (DCT)*, *Discrete Wavelet Transform (DWT)* [3]. Pada penelitian *Least Significant Bits (LSB)* metode ini banyak dilakukan dengan pendekatan yang sederhana menyisipkan suatu informasi pada suatu media dengan mengganti nilai-nilai bit dengan bit data yang ingin disisipkan. Namun teknik ini memiliki kelemahan jika sebuah pesan yang akan disisipkan jumlahnya besar maka hasil dari media yang telah disisipkan akan mengalami distorsi besar.

Salah satu metode algoritma yang akan digunakan pada penelitian ini adalah *Pixel Value Differencing* (*PVD*) dikarenakan metode ini dikembangkan untuk meningkatkan daya tampung pesan dan pengurangan tingkat distorsi pada steganografi[4]. Cara kerja teknik *Pixel Value Differencing* (*PVD*) adalah dengan cara membagi media yang akan disisipkan menjadi blok-blok piksel yang bertetangga[5]. Blok-blok tersebut terdiri dari dua buah *pixel* yang posisinya berdekatan[6]. Bit–bit pesan yang akan disisipkan dihitung dengan besarnya kedua piksel tersebut.

Berdasarkan kelebihan sistem operasi Android dan metode *Pixel Value Differencing (PVD)*, maka dalam penelitian ini akan dibuat Perancangan Aplikasi Steganografi Berbasis Android Dengan Metode *Pixel Value Differencing (PVD)* untuk proses penyisipan pesan dan ekstraksi pesan[7].

II. LANDASAN TEORI

A. *Citra Digital*

Citra *(image)* adalah gambar pada bidang dua dimensi. Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dua dimensi. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut[8].

Citra (*image*) adalah kombinasi antara titik, garis, bidang dan warna untuk menciptakan suatu imitasi dari suatu objek, biasanya objek fisik atau manusia. Citra bisa berwujud gambar (*picture*) dua dimensi seperti lukisan, foto, dan yang berwujud tiga dimensi seperti, patung[2]. Adapun format *file* gambar adalah :

a) Bitmap (BMP)
   1) Merupakan format citra yang baku dilingkungan sistem Microsoft Windows dan IBM OS/2.
   2) Kualitas BMP lebih baik dan dengan ukuran yang lebih baik dari format JPG/ JPEG dan GIF.
   3) Format *file* Bitmap versi baru dari Microsoft Windows, setiap berkas/*file* terdiri dari: *header file, header bitmap*, informasi palet, dan data bitmap.

b) *Joint Photographic Experts Group (JPEG)*
   1) Dikembangkan oleh para ahli fotografi untuk mendapatkan gambar yang berukuran rasional tapi tetap menyimpan persepsi gambar yang baik.
   2) Bersifat *lossy* dengan tingkat *lossness* yang dapat diatur.
   3) Bagus untuk mengkompresi foto-foto natural tetapi kurang cocok untuk *computer-generated images* (CGI).

c) *Portable Graphics Network (PNG)*
   1) PNG digunakan di internet dan merupakan format "pengganti" GIF, setelah GIF terkena patent LZW yang dilakukan oleh Unisys.
   2) Diprakarsai oleh Thomas Boutell dari PNG Development Group, dan versi finalnya di-*release* pada 1 Oktober 1996.
   3) Memiliki kedalaman warna 48-bit.

B. *Steganografi*

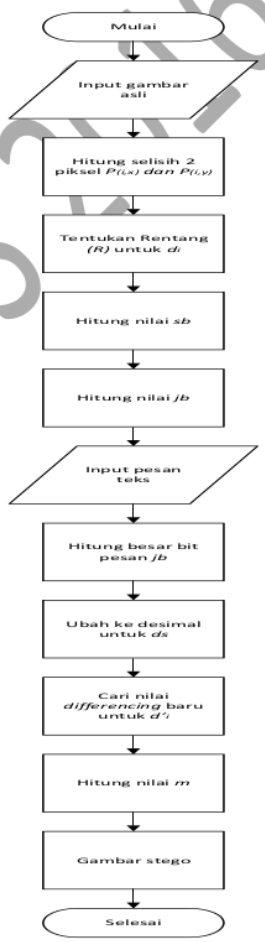
Steganografi *(steganography)* adalah ilmu dan seni menyembunyikan pesan rahasia *(hiding message)* sedemikian sehingga keberadaan pesan tidak terdeteksi oleh indera manusia[9]. Steganorafi berasal dari Bahasa Yunani yang berarti "tulisan tersembunyi" *(covered writing)*. Steganografi membutuhkan wadah penampung dan data rahasia yang akan disembunyikan [10].

1. Algoritma *Pixel Value Differencing (PVD)*

Berikut adalah aturan-aturan persamaan algoritma *Pixel Value Differencing* (PVD):

$$(P'_{(i,x)},(P'_{(i,y)}) = \begin{cases} (P_{(i,x)}+\lfloor m/2 \rfloor, P_{(i,y)}-\lfloor m/2 \rfloor), & \text{Jika } P_{(i,x)} \geq P_{(i,y)} \text{ dan } d'_i > d_i; \\ (P_{(i,x)}-\lfloor m/2 \rfloor, P_{(i,y)}+\lfloor m/2 \rfloor), & \text{Jika } P_{(i,x)} < P_{(i,y)} \text{ dan } d'_i > d_i; \\ (P_{(i,x)}-\lfloor m/2 \rfloor, P_{(i,y)}+\lfloor m/2 \rfloor), & \text{Jika } P_{(i,x)} \geq P_{(i,y)} \text{ dan } d'_i \leq d_i; \\ (P_{(i,x)}+\lfloor m/2 \rfloor, P_{(i,y)}-\lfloor m/2 \rfloor), & \text{Jika } P_{(i,x)} < P_{(i,y)} \text{ dan } d'_i \leq d_i; \end{cases}$$

Algoritma Penyisipan Pesan



Gambar 1. *Flowchart* Penyisipan

Algoritma Ekstraksi Pesan

Gambar 2. *Fowchart* Ekstraksi

2. *Peak Signal to Noise Ratio (PNSR)* dan *Mean Square Error (MSE)*

PSNR 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 1 memperlihatkan nilai PSNR beserta penjelasannya.

TABEL 1. 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	*Poor*, terdapat banyak derau pada citra
20 dB	*Unusable*

Berikut adalah sebuah tabel jangkauan yang digunakan untuk menentukan banyaknya bit yang akan disisipkan:

TABEL 2 BIT YANG DAPAT DISISIPKAN PADA DAERAH RENTANG

R (Rentang)	1	2	3	4	5	6
bb (batas bawah)	0	8	16	32	64	128
ba (batas atas)	7	15	31	63	127	255
jb (jumlah bit)	3	3	4	5	6	7

Pada penelitian ini, 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:

$$MSE = \frac{1}{MN} \sum_{y-1}^{M} \sum_{x-1}^{N} [I(x,y) - J(x,y)]^2 \quad (1)$$

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 maksimum sinyal dibagi dengan MSE.

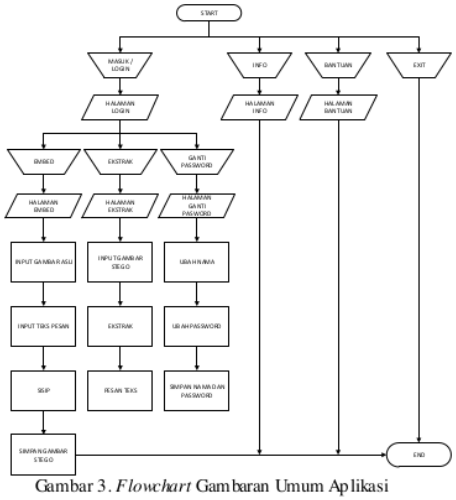
$$PSNR = 10 \; x \; log_{10} \left(\frac{MAXi}{MSE}\right) \quad (2)$$

Keterangan:
PSNR : nilai PSNR gambar (dalam dB)
MAXi : nilai maksimum piksel gambar
MSE : nilai MSE

III. PERANCANGAN SISTEM

Metode yang digunakan dalam perancangan sistem yaitu *flowchart* dan *storyboard* yang menggambarkan tentang proses yang terjadi pada sistem yaitu proses masuk *(login)*, penyisipan pesan *(embedding)*, ubah *password (update password)*, dan ekstrak pesan *(ekstract)*.

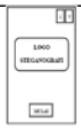
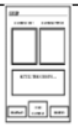
A. *Flowchart Gambaran Umum Aplikasi*

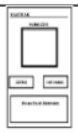


Gambar 3. *Flowchart* Gambaran Umum Aplikasi

B. Storyboard Aplikasi

TABEL 3 STORYBOARD APLIKASI ANDROID

No	Nama	Desain	Keterangan
1	*Splash Screen*		• Halaman ini menampilkan **Logo** aplikasi sebagai branding aplikasi. • Durasi 3-5 detik.
2	Halaman Awal		• Pada halaman awal terdapat **Logo** aplikasi di tengah, tombol **Mulai** di bawah, tombol **(i)** dan tombol **(?)** dipojok kanan atas. • Tombol **Mulai** untuk menuju ke halaman **Login** yang digunakan untuk menginput nama dan password pengguna. • Tombol **(i)** untuk menuju ke halaman **Info** yang menampilkan informasi/versi dari aplikasi. • Tombol **(?)** untuk menuju ke halaman **Bantuan** yang menampilkan isi petunjuk penggunaan aplikasi.
3	Halaman Info		• Pada pojok kiri atas terdapat judul halaman. • Halaman ini menampilkan versi dari aplikasi.
4	Halaman Bantuan		• Pada pojok kiri atas terdapat judul halaman. • Halaman ini berisi tentang petunjuk penggunaan aplikasi.
5	Halaman Login		• Pada pojok kiri atas terdapat judul halaman. • Dibagian tengah terdapat **Nama** dan **Password**. • Dibagian bawah terdapat tombol **Masuk**.
			• **Nama** digunakan untuk menginput nama pengguna. • **Password** digunakan untuk menginput password pengguna. • Tombol **Masuk** untuk memproses akun pengguna tersebut ke halaman **Menu**.
6	Halaman Menu		• Pada pojok kiri atas terdapat judul halaman. • Dibagian tengah terdapat **Logo** aplikasi. • Dibagian bawah terdapat tombol **Ekstrak**, tombol **Ganti Password** dan tombol **Sisip**. • Tombol **Ekstrak** untuk menuju ke halaman **Ekstrak.** • Tombol **Ganti Password** untuk menuju ke halaman **Ganti Password**. • Tombol **Sisip** untuk ke halaman **Sisip**.
7	Halaman Ganti Password		• Pada pojok kiri atas terdapat judul halaman. • Dibagian tengah terdapat **Nama lama, Nama baru, Password lama, Password baru** dan tombol **Ganti**. • **Nama lama** merupakan nama dari pengguna awal. • **Nama baru** digunakan untuk nama pengguna baru. • **Password lama** merupakan password dari pengguna awal. • **Password baru** digunakan untuk password pengguna baru. • Tombol **Ganti** untuk memproses/ mengupdate pengguna baru.
8	Halaman Sisip		• Pada pojok kiri atas terdapat judul halaman. • Dibagian tengah terdapat kotak **Gambar asli**, kotak **Gambar stego** dan kotak **Pesan.** • Dibagian bawah terdapat tombol **Sisip,** tombol **Cari Gambar** dan tombol **Simpan.** • Kotak **Gambar asli** menampilkan gambar asli yang akan digunakan untuk menyisipkan pesan. • Kotak **Gambar stego** menampilkan gambar stego hasil dari penyisipan pesan. • Kotak **Pesan** digunakan untuk mengisi pesan teks yang akan disisipkan ke gambar asli. • Tombol **Sisipkan** digunakan untuk memproses penyisipan pesan ke dalam gambar asli. • Tombol **Cari Gambar** digunakan untuk mencari gambar asli dari

			penyimpanan internal. • Tombol **Simpan** digunakan untuk menyimpan gambar stego ke penyimpanan internal.
9	Halaman Ekstrak		• Pada pojok kiri atas terdapat judul halaman. • Dibagian tengah terdapat kotak **Gambar Stego**, tombol **Ekstrak** dan tombol **Cari Gambar**. • Dibagian bawah terdapat kotak **Pesan** dari hasil ekstraksi. • Tombol **Cari Gambar** digunakan untuk mencari gambar stego yang akan di ekstraksi dalam penyimpanan internal. • Kotak **Gambar Stego** akan menampilkan gambar yang telah disisipkan pesan. • Tombol **Ekstrak** digunakan untuk proses ekstraksi pesan dari gambar stego. • Kotak **Pesan** akan menampilkan isi pesan hasil ekstraksi dari gambar stego.

TABEL 4. STORYBOARD APLIKASI MATLAB

No	Nama	Desain	Keterangan
1	Halaman Menu		• Halaman menu terdapat kotak **Gambar Asli**, **Gambar Stego**, **Psnr Mse**, dan tombol **Cari**. • Kotak **Gambar Asli** akan menampilkan gambar sebelum pesan disisipkan. • Kotak **Gambar Stego** akan menampilkan gambar yang telah disisipkan pesan. • Tombol **Cari** digunakan untuk mencari gambar asli atau gambar stego yang akan di hitung nilai psnr dan msenya. • Tombol **Hitung** digunakan untuk proses perhitungan nilai psnr dan mse dari gambar asli dan gambar stego dan kemudian hasilnya ditampilkan.

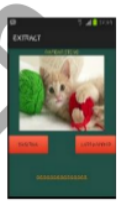
IV. IMPLEMENTASI DAN PENGUJIAN

Pengujian sistem ini dilakukan dengan pendekatan *Desain test case,* yaitu dengan menguji aplikasi dengan mencari fungsi–fungsi yang salah, kesalahan desain *interface*, kesalahan struktur berkas atau akses eksternal. *Black-box* berfokus pada persyaratakan fungsional perangkat lunak, sehingga mendapatkan serangkaian kondisi input yang seluruhnya menggunakan syarat fungsional pada suatu program *(MSE PSNR)*.

A. *Pengujian Aplikasi*

Gambar 4. Halaman *Splash Screen*

Gambar 5. Proses Penyisipan



Gambar 6. Proses Ekstraksi

B. *Pengujian Gambar*

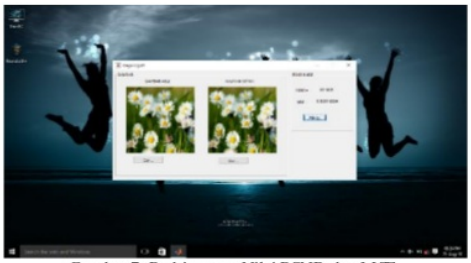
Pada penelitian ini terdapat enam buah gambar dengan beberapa tipe dan ukuran yang digunakan

TABEL 5. GAMBAR YANG AKAN DI UJI DENGAN UKURAN PIKSEL BERVARIASI

Nama Gambar	Gambar	Ukuran Gambar (piksel)	Tipe Gambar	Ukuran gambar (KB)
Bunga		128 x 128	PNG	42.7
Ferrari		128 x 128	JPG	10.7
Singa		128 x 128	BMP	48.0

C. *Pengujian Nilai MSE dan PSNR*

Algoritma MSE dan PSNR dibuat dengan menggunakan bahasa pemrograman Matlab. Matlab merupakan bahasa pemrograman yang sangat baik untuk mengolah gambar karena dilengkapi fungsi-fungsi yang memudahkan pemakaiannya. Berikut ini merupakan program yang digunakan untuk mengetahui nilai MSE dan PSNR dari setiap gambar yang sebelum dan sesudah disisipkan pesan



Gambar 7. Perhitungan Nilai PSNR dan MSE

TABEL 6. GAMBAR YANG AKAN DI UJI DENGAN UKURAN PIKSEL SAMA

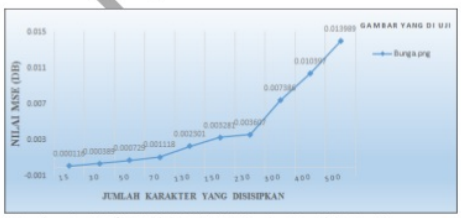
Nama Gambar	Gambar	Ukuran Gambar (piksel)	Tipe Gambar	Ukuran gambar (KB)
Bunga		512 x 512	PNG	734
Kucing		912 x 800	BMP	2000
Singa		128 x 116	JPG	10.9

D. *Hasil Pengujian*

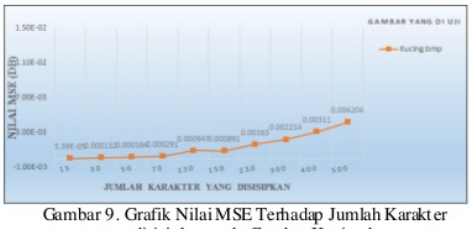
Hasil pengujian nilai MSE pada gambar dengan ukuran piksel dan tipe yang berbeda dapat dilihat pada Tabel 7.

TABEL 7. HASIL PENGUJIAN GAMBAR DALAM NILAI MSE (DB)

Jumlah karakter yang disisipkan	NILAI MSE (db)		
	Bunga.png	Kucing.bmp	Singa.jpg
15	0.000116	0.000053	1.20842
30	0.000389	0.000132	1.23682
50	0.000729	0.000184	1.32368
70	0.001118	0.000291	1.24524
130	0.002301	0.000947	1.29443
150	0.003281	0.000891	1.30758
230	0.003607	0.00163	1.42143
300	0.007386	0.002214	1.45317
400	0.010397	0.00311	1.55859
500	0.013989	0.004204	1.62022



Gambar 8. Grafik Nilai MSE Terhadap Jumlah Karakter yang disisipkan pada Gambar Bunga.png



Gambar 9. Grafik Nilai MSE Terhadap Jumlah Karakter yang disisipkan pada Gambar Kucing.bmp

Gambar 10. Grafik Nilai MSE Terhadap Jumlah Karakter yang disisipkan pada Gambar Singa.jpg

Hasil pengujian nilai PSNR pada gambar dengan ukuran piksel dan tipe yang berbeda dapat dilihat pada Tabel 8.

TABEL 8. HASIL PENGUJIAN GAMBAR DALAM NILAI PSNR (DB)

Jumlah karakter yang disisipkan	NILAI PSNR (db)		
	Bunga.png	Kucing.bmp	Singa.jpg
15	87.4835	90.848	47.3426
30	82.2642	86.9579	47.2417
50	79.5323	85.503	46.947
70	77.6766	83.5187	47.2123
130	74.5446	78.4009	47.044
150	73.0035	78.6621	47.0001
230	72.5928	76.0414	46.6376
300	69.4804	74.7121	46.5416
400	67.9955	73.236	46.2375
500	66.7067	71.9276	46.0691

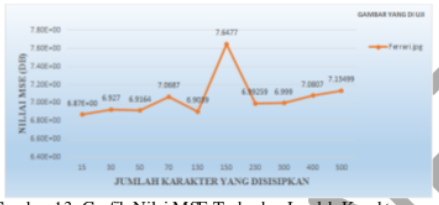
Gambar 11. Grafik Perbandingan Nilai PSNR Terhadap Jumlah Karakter yang disisipkan pada Gambar Bunga.png, Kucing.bmp dan Singa.jpg

Hasil pengujian nilai MSE pada ketiga gambar dengan ukuran piksel yang sama dapat dilihat pada Tabel 9.

TABEL 9. HASIL PENGUJIAN GAMBAR DALAM NILAI MSE (DB)

Jumlah karakter yang disisipkan	NILAI MSE (db)		
	Bunga.png	Ferrari.jpg	Singa.bmp
15	0.003031	6.8701	0.00921
30	0.006327	6.927	0.01747
50	0.013590	6.9164	0.03108
70	0.379171	7.0687	0.04317
130	0.39679	6.9039	0.08593
150	0.060852	7.6477	0.10424
230	0.069925	6.99259	0.16278
300	0.128906	6.999	0.22363
400	0.139201	7.0807	0.28945
500	0.223897	7.13499	0.52794

Gambar 12. Grafik Nilai MSE Terhadap Jumlah Karakter yang disisipkan pada Gambar Bunga.png



Gambar 13. Grafik Nilai MSE Terhadap Jumlah Karakter yang disisipkan pada Gambar Ferrari.jpg

Gambar 14. Grafik Nilai MSE Terhadap Jumlah Karakter yang disisipkan pada Gambar Singa.bmp

Hasil pengujian nilai PSNR pada ketiga gambar dengan ukuran piksel yang sama:

TABEL 10. HASIL PENGUJIAN GAMBAR DALAM NILAI PSNR (DB)

Jumlah karakter yang disisipkan	NILAI PSNR (db)		
	Bunga.png	Ferrari.jpg	Singa.bmp
15	73.3483	39.7951	68.5192
30	70.1526	39.7593	65.7403
50	66.834	39.766	63.239
70	52.3765	39.6714	61.8128
130	52.1792	39.7738	58.823
150	60.322	39.3295	57.9841
230	59.7184	39.7184	56.0488
300	57.0621	39.7144	54.6694
400	56.7284	39.664	53.5488
500	54.6643	39.6309	50.939

Gambar 15. Grafik Perbandingan Nilai PSNR Terhadap Jumlah Karakter yang disisipkan pada Gambar Bunga.png, Ferrari.jpg dan Singa.bmp

E. *Pengujian Stress Testing*

Stress *testing* adalah salah satu jenis pengujian sistem *(system testing)*. Pengujian ini bertujuan untuk melihat apakah perangkat lunak secara keseluruhan mampu menangani kebutuhan sumberdaya yang tidak normal (mencakup kuantitas, frekuensi, maupun *volume*). Apakah data dalam jumlah sangat besar, dengan frekuensi sangat tinggi, serta *volume* yang sangat besar mengakibatkan performa atau bahkan fungsionalitas perangkat lunak terganggu atau tidak. Jadi, meskipun perangkat lunak anda sudah berjalan baik di setiap fiturnya, pastikan juga setiap perangkat lunak tetap bekerja dengan maksimal saat diberi 'beban berat'. Jangan hanya mengu engan data uji yang terbatas [1].

Stress test dilakukan secara bertahap yaitu pertama dengan *test* kecil dan dilanjutkan dengan maksimal yang bisa ditangani oleh aplikasi, ini penting karena kita harus mengetahui seberapa besar aplikasi dalam menangani beban.

TABEL 11 HASIL PENGUJIAN STRESS TESTING

Nama Gambar	Gambar	Ukuran (piksel)	Tipe	Ukuran (KB)	Maks pesan (karakter)
Bunga		512 x 512	PNG	734	526484
Kucing		912 x 800	BMP	2080	1576067
Singa		128 x 116	JPG	10.9	8093
Bunga		128 x 128	PNG	42.7	32178
Ferrari		128 x 128	JPG	10.7	8992

Singa		128 x 128	BMP	48.0 KB	34244

V. KESIMPULAN DAN SARAN

A. *Kesimpulan*

1. Aplikasi steganografi dengan metode *Pixel Value Differencing (PVD)* pada *mobile phone* berbasis Android ini dapat menyembunyikan informasi dengan melakukan penyisipan pesan dan ekstraksi pesan rahasia.
2. Teknik steganografi dengan metode *Pixel Value Differencing (PVD)* dapat digunakan pada sebuah gambar dengan 3 tipe berbeda yaitu: .png, .jpg dan .bmp dengan ukuran yang berbeda.
3. Berdasarkan dari hasil analisa penelitian ini dapat disimpulkan bahwa:
   a. Aplikasi steganografi yang dibuat dari implementasi algoritma *Pixel Value Differencing (PVD)* menunjukkan bahwa kualitas terhadap ketiga gambar yang diujikan dengan ukuran dan tipe berbeda masih dalam keadaan baik yaitu berkualitas *reasonable* sampai dengan *excelent,* karena PSNR berkisar antara 46.0 db sampai 90.8 db yaitu menunjukan perubahan kualitas gambar rendah.
   b. Pada proses ekstraksi, pesan yang disisisipkan pada gambar dalam aplikasi steganografi ini, dapat diekstrak kembali dengan baik yaitu pesan yang disisipkan sebelum proses penyisipan dan setelah proses ekstraksi sama tanpa ada perubahan atau *error* yang menyebabkan isi pesan tidak dapat dikembalikan.
   c. Hasil pengujian nilai MSE dan PSNR terhadap gambar yang dihasilkan dari aplikasi steganografi ini pun menunjukkan nilai yang cukup baik bergantung pada tipe gambar yang digunakan dan besarnya jumlah karakter yang disisipkan pada gambar tersebut. Hasil pengujian pada gambar dengan ukuran berbeda, gambar Kucing.bmp memperoleh nilai desibel yang lebih baik antara 71.0 sampai 90.0 dibandingkan dengan gambar Bunga.png antara 66.0 sampai 87.0 dan gambar Singa.jpg antara 46.0 sampai 47.0. Ini menunjukkan bahwa diantara gambar Bunga.png dan gambar Singa.jpg, gambar Kucing.bmp lebih baik untuk penyisipan pesan rahasia dengan perubahan kualitas gambar sangat kecil karena memperoleh nilai PSNR yang tinggi, semakin besar nilai PSNR semakin kecil perubahan kualitas gambar.

B. *Saran*

Aplikasi ini bisa dikembangkan lagi untuk penelitian selanjutnya terutama untuk mengirimkan dan menerima pesan gambar hasil steganografi.

1. Untuk lebih mengamankan data/informasi yang sangat rahasia dapat menggunakan penggabungan antara teknik steganografi dan kriptografi.
2. Pengembangan selanjutnya dapat menggunakan algoritma steganografi lain yang dapat mengkompresi sebuah gambar dengan *input* dan *output* gambar yang mempunyai format lebih bervarian lagi.
3. Aplikasi ini hanya dapat menyisipkan sebuah pesan, sehingga untuk penelitian selanjutnya dapat menggunakan *file* dengan format/tipe lainnya *.doc, *.ppt, *.pdf dan sebagainya

REFERENSI

[1] Jayadi, "Kompasiana," 5 Januari 2011. [Online]. Available:http://www.kompasiana.com/highspeed/stress-test. [Accessed 1 Agustus 2016].

[2] R. Munir, "Steganografi dan Watermarking," Bahan Kuliah ke-7, 2004.

[3] Mehdi Hussain; Ainuddin Wahid Abdul Wahab; Nor Badrul Anuar; Rosli Salleh; Rafidah Md Noor, 2015, "Pixel value differencing steganography techniques: Analysis and open challenge", IEEE International Conference on Consumer Electronics - Taiwan, Pages: 21 – 22.

[4] Zhenhao Zhu; Tao Zhang; Pengwei Zhu; Baoji Wan; Xiaodan Hou, 2013, "Steganalysis of AE-LSB steganography based on pixel value differencing", Ninth International Conference on Natural Computation (ICNC), Pages: 1449 – 1453.

[5] C. Wu and W. H. Tsai, "A steganographic method for images by pixel-value differencing", vol. 24, pp. 1613-1626, 2003.

[6] K. Gulve Avinash; M. S. Joshi, 2012, "A Secured Five Pixel Pair Differencing Algorithm for Compressed Image Steganography", Third International Conference on Computer and Communication Technology, Pages: 278 – 282.

[7] Rozali, S. Guritman and H. T. Natalisa, "Perbaikan Dan Evaluasi Kinerja Algoritma Pixel- Value Differencing (PVD)", vol. 09, 2009.

[8] R. Munir, "Pengantar Pengolahan Citra". Bandung, 2004.

[9] W. T. Hsien and S. L. Hui, "A Steganographic Method Based on Pixel-Value Differencing and the Perfect Square Number," Journal of Applied Mathematics, vol. 2013, no. 5, pp. 8, 2013.

[10] M. A. Andriawan , S. and S. J. I. Ismail, "Implementasi Steganografi Pada Citra Digital File Gambar Bitmap (BMP) Menggunakan Java", Bandung: Politeknik Telkom.

Estu Sinduningrum - Perancangan Aplikasi Steganografi Berbasis Android dengan Metode Pixel Value Differencing (PVD)

ORIGINALITY REPORT

22%	21%	5%	13%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.gunadarma.ac.id Internet Source	3%
2	yaniwid.wordpress.com Internet Source	2%
3	Rojali, Ida Sri Rejeki Siahaan, Benfano Soewito. "Steganography algorithm multi pixel value differencing (MPVD) to increase message capacity and data security", AIP Publishing, 2017 Publication	2%
4	media.neliti.com Internet Source	2%
5	digilib.its.ac.id Internet Source	2%
6	Submitted to Universitas Dian Nuswantoro Student Paper	1%
7	library.binus.ac.id Internet Source	1%

8	repositori.uin-alauddin.ac.id Internet Source	1%
9	text-id.123dok.com Internet Source	1%
10	hurufba.blogspot.com Internet Source	1%
11	fr.scribd.com Internet Source	1%
12	e-journals.unmul.ac.id Internet Source	1%
13	Submitted to Universitas Sebelas Maret Student Paper	1%
14	www.ijert.org Internet Source	1%
15	eprints.unsri.ac.id Internet Source	1%
16	www.ijitee.org Internet Source	1%
17	eprints.unm.ac.id Internet Source	<1%

Exclude quotes	On	Exclude matches	< 17 words

Exclude bibliography On