Tersedia di https://jtsiskom.undip.ac.id (14 Maret 2018) Jurnal Teknologi dan Sistem Komputer, 6(2), 2018, 44-50
DOI: 10.14710/jtsiskom.6.2.2018.44-50

Aplikasi Steganografi Menggunakan LSB 4 Bit Sisipan dengan Kombinasi Algoritme Substitusi dan Vigenere Berbasis Android

Priyagung Hernawandra, Supriyadi*), U. Tresna Lenggana

Program Studi Informatika, STMIK Kharisma Karawang
Jl. Pangkal Perjuangan Km.1 Bypass, Karawang, Indonesia 41316

Cara sitasi: P. Hernawandra, S. Supriyadi, and U. T. Lenggana, "Aplikasi Steganografi Menggunakan LSB 4 Bit Sisipan dengan Kombinasi Algoritme Substitusi dan Vigenere Berbasis Android," Jurnal Teknologi dan Sistem Komputer, vol. 6, no. 2, Apr. 2018. doi: 10.14710/jtsiskom.6.2.2018.44-50, [Online].

***Abstract** – This study aims to implement steganography with 4 bit LSB method combined with Vigenere and substitution encryption algorithms. This combination strengthens the security of text messages that are inserted into the image because the message is already a ciphertext as the result of the Vigenere and substitution encryption process. This steganography is realized as an application that runs on Android devices. This application can insert text messages that contain space characters and a combination of uppercase letters in a digital image. Using this application, insertion of messages into images causes the increase of images size by an average of 12.77% of the original size from 10 sample images.*

***Keywords** – steganography; 4 bit LSB; substitution algorithm, Vigenere algorithm*

***Abstrak** – Penelitian ini bertujuan untuk mengaplikasikan steganografi dengan metode LSB 4 bit penyisipan yang digabungkan dengan algoritme enkripsi Vigenere dan substitusi. Penggunaan algoritme enkripsi ini memperkuat keamanan pesan teks yang disisipkan ke gambar karena pesan tersebut sudah berupa ciphertext hasil dari proses enkripsi dengan Vigenere dan substitusi. Pesan yang berisi karakter spasi dan kombinasi huruf besar kecil juga dapat disisipkan dalam citra digital. Steganografi ini diwujudkan sebagai aplikasi yang berjalan di devais Android. Dengan menggunakan aplikasi ini, penyisipan pesan dengan metode LSB 4 bit sisipan ke gambar menyebabkan penambahan besar gambar rata-rata 12,77% dari besar aslinya untuk 10 gambar uji.*

***Kata Kunci** – steganografi; LSB 4 bit sisipan; algoritme substitusi; algoritme Vigenere*

I. Pendahuluan

Steganografi digunakan untuk menyembunyikan pesan rahasia pada media lain sehingga keberadaan pesan tidak terdeteksi orang lain selain penerima pesan. Pemilihan media penampung perlu diperhatikan karena penyisipan pesan akan berpengaruh pada kualitas dan ukuran media yang digunakan [1]. Media yang dapat digunakan adalah berupa teks, citra, audio maupun video [2].

Steganografi ke media gambar dapat dilakukan dengan menyisipkan pesan ke digit biner paling lemah dalam sebuah piksel, yang disebut metode *least significant bit (LSB)* [3]-[6]. Steganografi dengan satu bit penyisipan menghasilkan matriks piksel gambar yang tidak berubah secara signifikan karena hanya bit terakhir yang diubah [5]. Penyisipan dua bit pesan dalam steganografi juga tidak mengubah kualitas gambar asli dengan gambar setelah disisipkan, walaupun mengubah ukuran gambarnya [4], [6]. LSB dengan tiga bit penyisipan juga menghasilkan kualitas gambar yang masih sama hanya saja ukurannya menjadi lebih besar dari gambar aslinya [1].

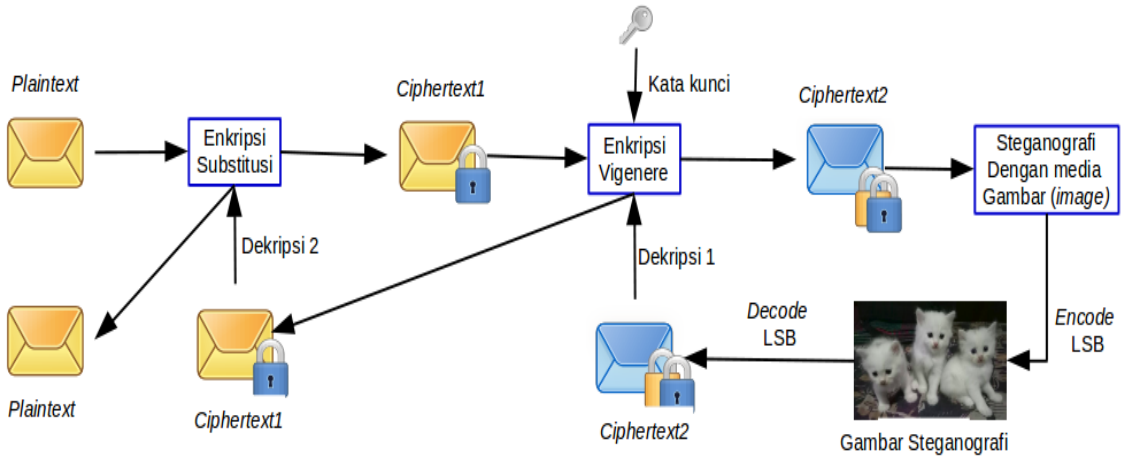
Steganografi dengan metode LSB 1 bit dapat mendeteksi posisi 1 bit penyisipan pesan yang mungkin disisipi dengan pola tertentu (berurutan atau acak) menggunakan metode *enhanced* LSB, misalnya dalam [7]. Apabila pesan yang disisipkan merupakan pesan asli, maka pesan akan mudah diketahui apabila bit yang mungkin itu diketahui. Untuk meningkatkan keamanan pesan tersebut, maka perlu ditambahkan enkripsi untuk mengubah pesan menjadi informasi acak (*chipertext*) yang dapat dibuka (dekripsi) menggunakan algoritme dan kunci tertentu [5].

Beragam algoritme enkripsi untuk meningkatkan keamanan dalam metode LSB steganografi telah diterapkan, di antaranya algoritme *playfair* [8], Vigenere [9] dan gabungan Vigenere RC4 [10]. Penggunaan algoritme *playfair* dalam [8] masih memiliki kekurangan pada pembentukan bitgram *plaintext* yang harus berpasangan. Apabila terdapat karakter tanpa adanya pasangan, maka harus ditambah karakter lain sehingga mengubah *plaintext* asli. Algoritme Vigenere dalam [9] memiliki masalah jika dalam pesan *plaintext* terdapat karakter spasi dan hanya huruf kecil yang dapat digunakan dalam pesan. Gabungan algoritme Vigenere dan RC4 dalam [10] memiliki kelemahan pada panjang pesan, yaitu karena hanya satu bit penyisipan.

Dari hal tersebut di atas, penelitian ini bertujuan mengaplikasikan steganografi menggunakan metode LSB 4 bit penyisipan yang dikombinasikan dengan algoritme enkripsi Vigenere dan substitusi untuk meningkatkan keamanan pesan. Pengkodean LSB

*) Penulis korespondensi (Supriyadi)
Email: fnfcreator@stmik-kharisma.ac.id)

Dikirim: 16 Desember 2017; Direvisi: 29 Desember 2017, 3 Februari 2018; Diterima: 14 Februari 2018



Gambar 1. Skema gabungan steganografi dan kriptografi

dengan 4-bit penyisipan memungkinkan gambar dapat menampung pesan lebih panjang, yaitu 2 piksel gambar 8 bit untuk menampung 1 karakter *American Standard Code for Information Interchange* (ASCII) 8 bit. Algoritme substitusi diterapkan untuk menghasilkan *ciphertext* sebagai masukan ke enkripsi Vigenere sehingga selain untuk meningkatkan keamanan pesan, juga akan mampu menampung pesan yang berisi karakter spasi dan kombinasi huruf besar kecil. Aplikasi steganografi dikembangkan untuk berjalan di devais *mobile* berbasis Android.

II. Metode Penelitian

Penelitian ini menggabungkan steganografi dan kriptografi untuk menyisipkan pesan teks ke media gambar menggunakan metode LSB 4 bit penyisipan, enkripsi Vigenere dan substitusi. Skema yang diusulkan dinyatakan dalam Gambar 1. Pesan teks (*plaintext*) akan dienkripsi menggunakan algoritme substitusi menghasilkan *Ciphertext1*. Hasil enkripsi ini dienkripsi kembali menggunakan Vigenere dengan kunci tertentu menghasilkan *Ciphertext2*. Hasil dari *Ciphertext2* dikonversi menjadi kode biner ASCII 8 bit yang disisipkan ke gambar dengan metode LSB.

Proses memperoleh pesan asli (dekripsi) dilakukan dengan dengan mengkonversi pesan pada gambar menjadi *Ciphertext2*. Setelah *Ciphertext2* diperoleh, langkah selanjutnya adalah didekripsi dengan dekriptor Vigenere sehingga menjadi *Ciphertext1*. Tahap akhir adalah menampilkan pesan asli, yaitu dengan cara mendekripsi *Ciphertext1* dengan dekriptor substitusi.

Penyandian pesan dalam algoritme substitusi menggunakan Persamaan 1. Penyandian dengan algoritme Vigenere menggunakan Persamaan 2. Kunci yang digunakan dalam algoritme Vigenere adalah "kerawang".

$$C_i = P_i + N \bmod 128 \quad (1)$$

$$C_i = P_i + K_i \bmod 128 \quad (2)$$

Keterangan:
C_i = *ciphertext* ke-i
N = panjang pesan asli
P_i = *plaintext* ke-i hasil enkripsi
K_i = kunci ke-i
mod 128 = 128 bit dari kode ASCII

Tabel 1. Kode karakter dalam pesan asli

Indeks ke-i	Pesan (P)	Kode ASCII	Indeks ke-i	Pesan (P)	Kode ASCII
P_1	S	83	P_{10}	r	114
P_2	T	84	P_{11}	i	105
P_3	M	77	P_{12}	s	115
P_4	I	73	P_{13}	m	109
P_5	K	75	P_{14}	a	97
P_6	(spasi)	32	P_{15}	(spasi)	32
P_7	K	75	P_{16}	9	57
P_8	h	104	P_{17}	2	50
P_9	a	97			

Proses dekripsi algoritme Vigenere menggunakan Persamaan 3. Dekriptor Vigenere ini menggunakan kunci yang sama seperti pada enkripsi yaitu “karawang”. Dekripsi algoritme substitusi menggunakan Persamaan 4.

$$P_i = C_i - K_i \bmod 128 \quad (3)$$

$$P_i = C_i - N \bmod 128 \quad (4)$$

Keterangan:
P_i = *plaintext* ke-i hasil dekripsi
C_i = *ciphertext* ke-i
N = panjang pesan
K_i = kunci ke-i
mod 128 = 128 bit dari kode ASCII

Aplikasi Android dikembangkan dengan menggunakan SDK Android versi 19 untuk menghasilkan aplikasi yang berjalan di Android 4.4. Pengembangan dilakukan di sistem operasi Linux Ubuntu 16.04.2 32 bit menggunakan IDE Eclipse dan JDK 1.8.

III. Hasil Dan Pembahasan

Berikut ini adalah analisis proses steganografi dan kriptografi yang ditunjukkan dalam Gambar 1. Pesan asli yang akan disisipkan ke dalam gambar adalah "STMIK Kharisma 92" yang mengandung spasi, huruf besar dan angka. Kunci simetris yang digunakan dalam enkripsi dan dekripsi adalah "karawang". Pesan asli dan kunci tersebut masing-masing ditransformasikan berdasarkan kode ASCII 7 bit (dinyatakan dalam desimal) seperti pada Tabel 1 dan Tabel 2.

Tabel 2. Kode karakter dalam kunci (K)

Indeks ke-i	Kunci (K)	Kode ASCII	Indeks ke-i	Kunci (K)	Kode ASCII
K_1	k	107	K_5	w	119
K_2	a	97	K_6	a	97
K_3	r	114	K_7	n	110
K_4	a	97	K_8	g	103

Tabel 3. Hasil enkripsi subtitusi

P_i	ASCII	C_i	*Ciphertext*
S	83	100	d
T	84	101	e
M	77	94	^
I	73	90	Z
K	75	92	\
(spasi)	32	49	1
K	75	92	\
h	104	121	y
a	97	114	r
r	114	131	f
i	105	122	z
s	115	132	“
m	109	126	~
a	97	114	r
(spasi)	32	49	1
9	57	74	J
2	50	67	C

Pesan yang disisipkan ke dalam gambar disandikan dengan algoritme substitusi menggunakan Persamaan 1. Proses enkripsi dilakukan menggunakan perhitungan enkripsi substitusi untuk semua karakter, dari P_1 = S sampai karakter terakhir yaitu P_{17} = 2. Hasil enkripsi ditunjukkan dalam Tabel 3. Pesan hasil enkripsi substitusi (*Ciphertext1*) adalah **de^Z\1\yr z ~r1JC**.

Pengacakan pesan tahap kedua algoritme Vigenere. Enkripsi Vigenere menggabungkan *plaintext* (*Ciphertext1*) dengan kunci K sehingga menghasilkan *ciphertext* yang baru (*Ciphertext2*) menggunakan Persamaan 2. Proses enkripsi dilakukan untuk semua karakter dari karakter pertama P_1 = d dan kunci K_1 = k sampai karakter terakhir yaitu P_{17} = 2 dan perulangan kunci K_{17} = k. Hasil enkripsi dengan algoritme Vigenere dinyatakan dalam Tabel 4. Pesan hasil enkripsi Vigenere (*Ciphertext2*) adalah **OFP;S J`]dleuS 1**.

Gambar yang digunakan untuk media steganografi penyisipan *Ciphertext2* adalah citra Kucing.png ukuran 781x504 piksel. Cuplikan awal kode biner gambar asli tersebut dipisahkan empat bit terakhirnya (dituliskan tebal) menjadi sebagai berikut:

0000**0000** 0000**0000** 0000**0000** 0000**0000**
0000**0000** 0000**0000** 0000**0000** 0000**0000**
0000**0000** 0000**0000** 0000**0000** 0000**0000**
0000**0000** 0000**0000** 0000**0000** 0000**0000**
0000**0000** 0000**0000** 0000**0000** 0000**0000**
0000**0000** 0000**0000** 0000**0000** 0000**0000**
0000**0000** 0000**0000** 0000**0000** 0000**0000**
0000**0000** 0000**0000** 0000**0000** 0000**0000**

Tabel 4. Hasil enkripsi dengan algoritme Vigenere

P_i	Pesan ASCII	K	Kunci ASCII	C_i	*Ciphertext*
d	100	k	107	79	O
e	101	a	97	70	F
^	94	r	114	80	P
Z	90	a	97	59	;
\	92	w	119	83	S
1	49	a	97	18	(null)
\	92	n	110	74	J
y	121	g	103	96	`
r	114	k	107	93	]
f	131	a	97	100	d
z	122	r	114	108	l
„	132	a	97	101	e
~	126	w	119	117	u
r	114	a	97	83	S
1	49	n	110	31	(null)
J	74	g	103	49	1
C	67	k	107	46	.

Tabel 5. Operasi logika XOR LSB 4 bit gambar dan karakter dan hasilnya (penyisipan pesan)

Karakter	Kode biner karakter	LSB 4 bit gambar	Hasil operasi XOR
O	0100 1111	0000 0000	**0100 1111**
F	0100 0110	0000 0000	**0100 0110**
P	0101 0000	0000 0000	**0101 0000**
;	0011 1011	0000 0000	**0011 1011**
S	0101 0011	0000 0000	**0101 0011**
(null)	0001 0010	0000 0000	**0001 0010**
J	0100 1010	0000 0000	**0100 1010**
`	0110 0000	0000 0000	**0110 0000**
]	0101 1101	0000 0000	**0101 1101**
d	0110 0100	0000 0000	**0110 0100**
l	0110 1100	0000 0000	**0110 1100**
e	0110 0101	0000 0000	**0110 0101**
u	0111 0101	0000 0000	**0111 0101**
S	0101 0011	0000 0000	**0101 0011**
(null)	0001 1111	0000 0000	**0001 1111**
1	0011 0001	0000 0000	**0011 0001**
.	0010 1110	0000 0000	**0010 1110**

0000**0000** 0000**0000** 00000000 00000000
00110000 00000000 00000000 00000000
00000000 00000000 00000000 00001101
11111111 11100000 01100000 00000000
00000000 00000000 00000000 01111111
11111111 00110000 .....

Pesan *Ciphertext2* dalam Tabel 4 diubah dalam kode biner dan disisipkan ke dalam gambar dengan operasi logika XOR. Dengan pesan *Ciphertext2* adalah **OFP;S J`]dleuS 1**, maka proses dan hasil operasi logika XOR-nya dengan kode LSB 4 bit dari 2 buah piksel gambar dinyatakan dalam Tabel 5. Setelah semua karakter disisipkan, perbedaan antara gambar sebelum dan setelah disisipkan ditunjukkan dalam Gambar 2. Secara kasat mata tidak ada perbedaan gambar sebelum disisipi

(a)

Sebelum penyisipan pesan

(b) Setelah penyisipan pesan

Gambar 2. Citra sebagai bahan uji teknik

Tabel 6. Operasi logika XOR LSB 4 bit gambar stego dan gambar asli

2 Kode LSB 4 bit Gambar		Hasil dekoding (XOR)	Kode Pesan
Stego	Asli		
0100 1111	0000 0000	0100 1111	**O**
0100 0110	0000 0000	0100 0110	**F**
0101 0000	0000 0000	0101 0000	**P**
0011 1011	0000 0000	0011 1011	**;**
0101 0011	0000 0000	0101 0011	**S**
0001 0010	0000 0000	0001 0010	**(null)**
0100 1010	0000 0000	0100 1010	**J**
0110 0000	0000 0000	0110 0000	**`**
0101 1101	0000 0000	0101 1101	**]**
0110 0100	0000 0000	0110 0100	**d**
0110 1100	0000 0000	0110 1100	**l**
0110 0101	0000 0000	0110 0101	**e**
0111 0101	0000 0000	0111 0101	**u**
0101 0011	0000 0000	0101 0011	**S**
0001 1111	0000 0000	0001 1111	**(null)**
0011 0001	0000 0000	0011 0001	**1**
0010 1110	0000 0000	0010 1110	**.**

pesan dengan gambar yang sudah disisipi gambar. Hal tersebut menunjukkan proses steganografi berhasil karena tidak menimbulkan kecurigaan. Kode biner gambar hasil Kucing.png yang sudah disisipi pesan steganografi (pesan terenkripsi dinyatakan dalam karakter tebal) adalah sebagai berikut:

0000**0100** 0000**1111** 0000**0100** 0000**0110** 0000**0101**
0000**0000** 0000**0011** 0000**1011** 0000**0101** 0000**0011**
0000**0001** 0000**0010** 0000**0100** 0000**1010** 0000**0110**
0000**0000** 0000**0101** 0000**1101** 0000**0110** 0000**0100**
0000**0110** 0000**1100** 0000**0110** 0000**0101** 0000**1111**
0000**0101** 0000**1101** 0000**0011** 0000**1001** 0000**1111**
0000**0011** 0000**0001** 0000**0010** 0000**1110** 00000000
00000000 00110000 00000000 00000000 00000000
00000000 00000000 00000000 00001101 11111111
11100000 01100000 00000000 00000000 00000000
00000000 01111111 11111111 00110000 .....

Tabel 7. Hasil dekripsi *Ciphertext2* dengan Vigenere

C_i	Desimal ASCII	K_i	Desimal ASCII	P_i	*Plaintext (P)*
O	79	k	107	100	d
F	70	a	97	101	e
P	80	r	114	94	^
;	59	a	97	90	Z
S	83	w	119	92	\
(null)	18	a	97	49	1
J	74	n	110	92	\
`	96	g	103	121	y
]	93	k	107	114	r
d	100	a	97	3	(null)
l	108	r	114	122	z
e	101	a	97	4	(null)
u	117	w	119	126	~
S	83	a	97	114	r
(null)	31	n	110	49	1
1	49	g	103	74	J
.	46	k	107	67	C

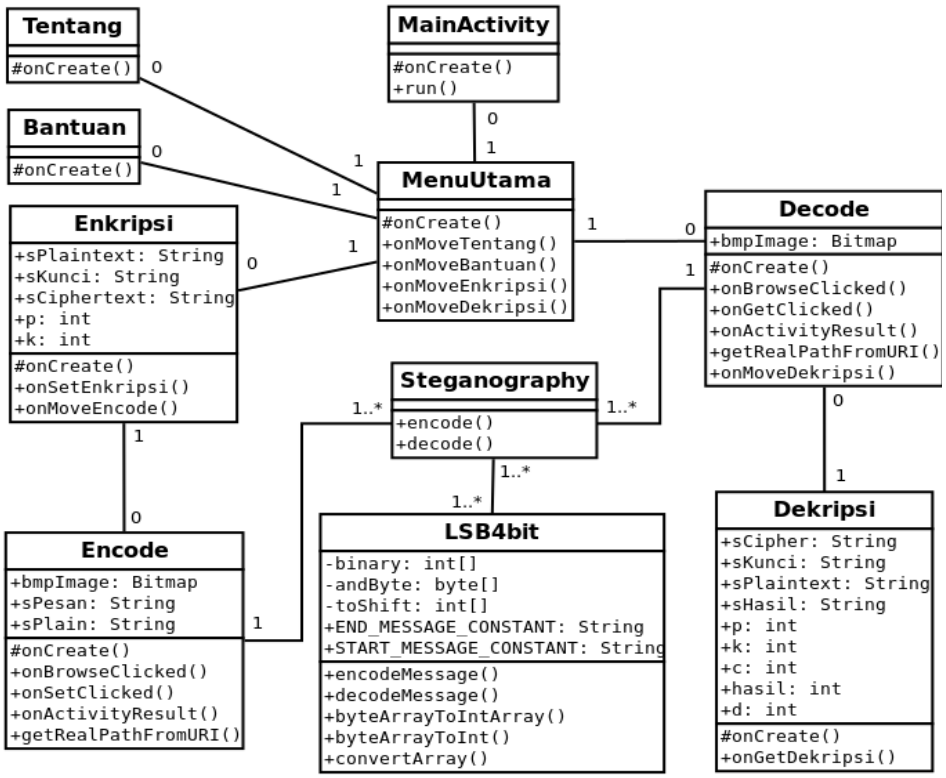
Tabel 8. Hasil dekripsi *Ciphertext1* dengan subtitusi

Plaintext (P)	Desimal ASCII	Desimal Dekripsi	*Plaintext (P)*
d	100	83	S
e	101	84	T
^	94	77	M
Z	90	73	I
\	92	75	K
1	49	32	(spasi)
\	92	75	K
y	121	104	h
r	114	97	a
(null)	3	114	r
z	122	105	i
(null)	4	115	s
~	126	109	m
r	114	97	a
1	49	32	(spasi)
J	74	57	9
C	67	50	2

Gambar yang telah disisipi pesan *Ciphertext2* disimpan pada galeri penyimpanan gambar. Untuk mendapatkan pesan dari Gambar steganografi yang telah tersimpan, maka dilakukan dekoding dengan meng-XOR-kan kode LSB 4 bit gambar tersisipi pesan dengan gambar asli seperti ditunjukkan dalam Tabel 6. Proses dekoding menghasilkan pesan tersisip, yaitu **OFP;S J`]dleuS 1**, yang merupakan *Ciphertext2* dari algoritme Vigenere**.**

Dekripsi *Ciphertext2* dengan algoritme Vigenere menggunakan Persamaan 3 dan kunci yang sama pada enkripsi yaitu "karawang". Hasil dekripsi *Cipthertext2* dinyatakan dalam Tabel 7, yang menghasilkan pesan *Ciphertext1* **de^Z\1\yr z ~r1JC.** Pesan hasil dekripsi Vigenere ini, didekripsi dengan algoritme substitusi untuk memperoleh pesan asli menggunakan Persamaan 4. Pesan hasil dekripsi substitusi adalah **STMIK Kharisma 92**.

Aplikasi steganografi dengan LSB 4 bit serta algoritme Vigenere dan substitusi dirancang untuk berjalan di sistem operasi Android. Aktor pada aplikasi ini adalah pengguna dengan melakukan aktivitas yang



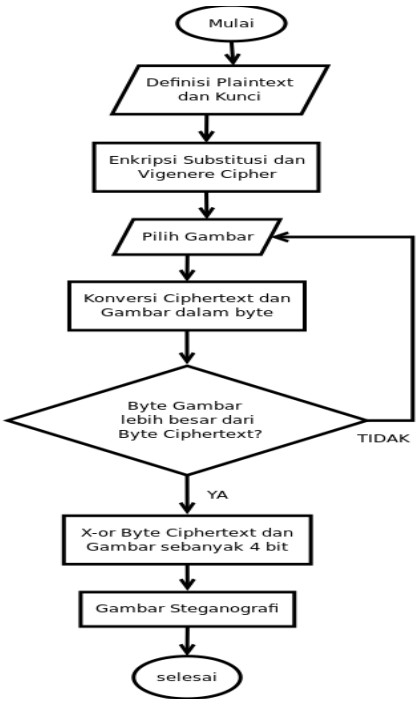
Gambar 3. Kelas diagram dalam aplikasi Android steganografi

Tabel 9. Deskripsi usecase dalam aplikasi steganografi

No	*Usecase*	Deskripsi
1.	*Splash*	Menampilkan *splash screen* aplikasi steganografi
2.	Menu Utama	Menampilkan menu aplikasi steganografi
3.	Enkripsi	Proses mengubah pesan asli menjadi pesan tersandi algoritme substitusi yang digabungkan dengan algoritme Vigenere *cipher*
4.	*Encode*	Proses penyisipan pesan tersandi pada gambar menggunakan metode LSB
5.	*Decode*	Proses pengambilan pesan dari gambar yang berupa pesan tersandi menggunakan metode LSB
6.	Dekripsi	Proses mengubah pesan tersandi menjadi pesan asli menggunakan algoritme Vigenere *cipher* dan algoritme substitusi
7.	Bantuan	Penjelasan tentang penggunaan aplikasi

terdiri dari tujuh *usecase* yang diuraikan pada Tabel 9. Aplikasi terdiri dari sepuluh kelas seperti ditunjukkan dalam diagram kelas pada Gambar 3.

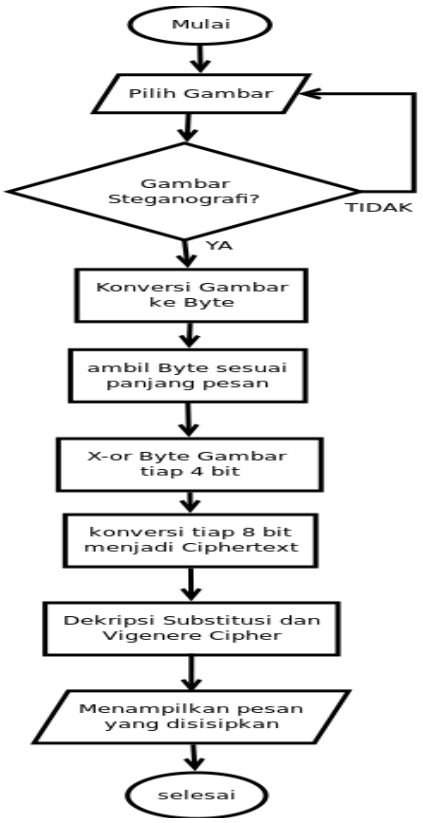
Proses pengkodean pesan menggunakan teknik kriptografi dan steganografi ditunjukkan dalam Gambar 4. Byte gambar yang dipilih harus lebih besar daripada byte *ciphertext* yang disisipkan dalam gambar. Diagram alir proses dekoding dan dekripsi pesan untuk mengembalikan pesan terkode yang tersimpan dalam gambar agar kembali ke pesan asli ditunjukkan dalam Gambar 5. Gambar yang dipilih diperiksa terlebih



Gambar 4. Diagram alir proses *encoding*

dahulu apakah merupakan gambar steganografi atau bukan sebelum dilakukan dekoding dan didekripsi.

Antar muka aplikasi Android dirancang agar dapat digunakan semudah mungkin dan seluruh proses enkoding dan dekoding pesan dapat dilakukan dengan baik. Elemen utama antarmuka aplikasi ini adalah tombol, label, layar gambar dan layar teks. Implementasi halaman enkripsi dan dekripsi pesan



Gambar 5. Diagram alir proses dekoding

Gambar 6. Tampilan halaman enkripsi dan dekripsi

Gambar 7. Tampilan halaman enkoding LSB 4 bit

menggunakan Vegenere dan substitusi ditunjukkan dalam Gambar 6. Tampilan halaman enkoding pesan tersandi ke gambar yang dipilih ditunjukkan dalam Gambar 7.

Penyisipan sebanyak 17 karakter, yaitu "STMIK Kharisma 92", pada berbagai gambar menghasilkan ukuran awal dan akhir yang berbeda. Pada penelitian ini,

Tabel 10. Hasil pengujian pengaruh penyisipan pesan terhadap besar gambar

No	Nama *File* Gambar	Ukuran Awal (KB)	Ukuran Akhir (KB)	Perbedaan Ukuran (%)
1	Kunci.png	479.4	382.1	-20.30%
2	Android.png	34.6	29.7	-14.16%
3	Buku.png	154.9	135.5	-12.52%
4	Melati.png	221	205.5	-7.01%
5	Kaos.png	78.4	76.1	-2.93%
6	Kertas.png	42.7	42.8	0.23%
7	Lampu.png	15.7	15.8	0.64%
8	Panda.png	247.2	249.9	1.09%
9	Kucing.png	329.2	376.9	14.49%
10	Jam.png	6.3	16.9	168.25%

10 gambar dengan format PNG digunakan sebagai media untuk menyisipkan pesan teks. Pengujian pengaruh penyisipan pesan ke beragam media gambar menggunakan LSB 4 bit serta enkripsi Vigenere dan substitusi ditunjukkan dalam Tabel 10. Sebagian besar gambar yang disisipkan pesan mengalami perubahan ukuran. Perubahan ukuran tersebut memiliki nilai rata-rata 12,77% dari ukuran gambar semula. Perbedaan ukuran tersebut tidak semuanya menghasilkan gambar akhir yang berukuran lebih besar, tetapi ada yang memiliki ukuran akhir lebih kecil dari ukuran semula. Hal tersebut diakibatkan oleh variasi faktor kompresi untuk masing-masing gambar hasil enkoding.

Penelitian ini telah mengimplementasikan steganografi untuk menyisipkan pesan teks ke media gambar menggunakan metode LSB. Gambar yang dihasilkan dan berisi pesan secara kasat mata tidak berbeda dari gambar aslinya (Gambar 2), seperti dinyatakan dalam [1], [3]-[6]. Dengan menggunakan LSB dari gambar sebesar 4 bit tiap piksel, metode LSB 4 bit penyisipan dalam penelitian ini akan mampu menampung lebih banyak pesan daripada LSB dengan penyisipan 1 bit [5], [10], 2 bit [4], [6] dan 3 bit penyisipan [1]. Dengan LSB 4 bit ini maka 1 karakter pesan dengan format ASCII dapat dikodekan dalam 2 piksel gambar 8 bit.

Sebelum dilakukan enkoding pesan ke gambar, pesan asli telah berhasil dienkripsi dengan algoritme Vigenere dan substitusi. Pesan yang dienkripsi ini tidak akan mudah dipahami jika tidak mempunyai kuncinya. Pesan asli yang dikirimkan tidak perlu diubah seperti halnya algoritme *playfair* [8] jika bitgram tidak mempunyai pasangan. Pesan yang dikirimkan dalam penelitian ini juga dapat mengandung karakter spasi dan huruf besar seperti halnya dalam [10] untuk mengatasi keterbatasan dalam [9] yang tidak mampu mengirimkan karakter tersebut.

IV. Kesimpulan

Aplikasi Android steganografi menggunakan metode LSB 4 bit yang digabungkan dengan algoritme substitusi dan Vigenere telah mampu menyisipkan pesan teks tersandi ke dalam media gambar. Aplikasi ini mampu menyisipkan pesan teks asli yang mengandung semua karakter ASCII 7 bit. Dari 10 gambar uji, penyisipan

pesan menyebabkan penambahan ukuran rata-rata sebesar 12.77% dari ukuran gambar awal.

Daftar Pustaka

[1] F. A. Antika, A. B. Purba, dan U. T. Lenggana. "Aplikasi Steganografi Pada Citra Digital Menggunakan Metode Least Significant Bit 3-3-2 Bit Berbasis Android," Skripsi, STMIK Kharisma Karawang, 2016.

[2] S. M. Rump, "Solving Nonlinear Systems with Least Significant Bit Accuracy," *Computing,* vol. 29, no. 3, pp. 182-200, 1982.

[3] F. Arifiansyah, N. Suciati, dan A. S. Wijaya, "Implementasi Boosted Steganography Scheme dengan Praposes Citra Menggunakan Histogram Equalization," Skripsi, Institut Teknologi Sepuluh Nopember, 2012.

[4] B. S. Champakamala, K. Padmini, and D. K. Radhika, "Least Significant Bit algorithm for image steganography," *International journal of Advanced Computer Technology*, vol. 3, no. 4, pp. 34-38, 2014.

[5] H. Manurung, "Teknik Penyembunyian Pesan Teks pada Media Citra GIF Dengan Metode *Least Significant Bi*t (LSB)," *Jurnal Pelita Informatika Budi Darma*, vol. 7, no. 2, pp. 62-68, 2014.

[6] Mujiono, Supriyadi, dan U. T. Lenggana, "Steganografi dengan Metode LSB 2-Bit pada citra JPEG Berbasis Android," Skripsi, STMIK Kharisma Karawang, 2014.

[7] E. R. Hidayat, dan K. Hastuti. "Analisis Steganografi Metode Least Significant Bit (LSB) dengan Penyisipan Sekuensial dan Acak secara Kuantitatif dan Visual," *Jurnal Techno.Com*, vol. 12, no. 3, pp. 157-167 , 2013.

[8] D. S. Anggaeni, U. T. Lenggana, dan A. B. Purba. "Implementasi Enkripsi Playfair Cipher ke dalam Steganografi dengan Metode Least Significant Bit (LSB)," Skripsi, STMIK Kharisma Kerawang, 2013.

[9] M. F. Syawal, D. C. Fikriansyah, dan N. Agani, "Implementasi Teknik Steganografi Menggunakan Algoritma Vigenere Cipher dan Metode LSB," *Jurnal TICom*, vol. 4, no. 3, pp. 91-99, 2016.

[10] B. Rakhmat, dan M. Fairuzabadi, "Steganografi Menggunakan Metode Least Significant Bit dengan Kombinasi Algoritma Kriptografi Vigenère dan RC4," *Jurnal Dinamika Informatika*, vol. 5, no. 2, pp. 1-17, 2010.