ISSN 2085-4552

Implementasi Steganografi Menggunakan Metode Least Significant Bit dan Kriptografi Advanced Encryption Standard

Shinta Puspita Sari, Winarno, Dodick Z. Sudirman

Program Studi Teknik Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia
shintapuspita@hotmail.com

Diterima 03 Juni 2012
Disetujui 12 Juni 2012

Abstract—Nowadays, the development of celluler device especially mobile phone is very rapid. The security of information in mobile phone becoming such an important thing, because users do not want other people can access their essential information. It means in the security in mobile phone. One way to secure the data is steganograpgy technique. This technique hides the data in a digital media. In this research the steganography technique that the is used is the Least Significant Bit (LSB) which implemented to cellular equipment with android as the operating system. The digital image which is used to hide the data is PNG. The test result shows that steganography has been implemented successfully in android phone. In the imprecebility aspect, user cannot different between the picture with message and the original picture bare eyedly. In recovery aspect, the message successfully encrypted and decrypted without changing the original content of the message. The lack occurs in fidelity aspect which the expansion of PNG file happens.

***Index Terms*—LSB, least significant bit, png, android, handphone**

I. PENDAHULUAN

Perkembangan Teknologi Informasi pada saat ini begitu pesat, diikuti pula dengan berkembangnya teknologi komunikasi seperti perangkat seluler yang biasa disebut dengan ponsel atau *handphone*. Hal ini dapat ditunjukkan dengan hasil survey Nielsen, bahwa jumlah pengguna ponsel per Mei 2011 mencapai 125 juta orang dari 238 juta penduduk di Indonesia.

Salah satu aspek yang sangat penting dalam komunikasi data adalah masalah keamanan dan kerahasiaan data. Dimana kebenaran dan keaslian suatu informasi sangat penting baik pada saat pengiriman ataupun pada saat informasi tersebut diterima (Pradana, 2011). Karena apabila informasi jatuh ke pihak lain, hal tersebut dapat menimbulkan kerugian bagi si pemilik informasi tersebut. Untuk itu diperlukan adanya cara atau teknik untuk mengamankan data atau informasi (Pratama, 2011).

Terdapat teknik yang digunakan untuk mengamankan dan menjaga kerahasian data, yaitu kriptografi dan steganografi. Kriptografi merupakan salah satu metode pengamanan data yang bertujuan digunakan untuk menjaga kerahasiaan data, keaslian data serta originalitas (Tumanggor, 2009). Sedangkan, steganografi adalah menyembunyikan informasi ke dalam sebuah media, bisa berupa media gambar, suara ataupun video (Firmansyah, 2011). Dengan demikian, dapat disimpulkan bahwa kriptografi fokus pada bagaimana melindungi isi informasi agar tetap aman (*secure*) dan steganografi fokus pada bagaimana agar isi informasi tersebut tidak terlihat keberadaannya (Prihanto, 2010).

Pada penelitian ini teknik kriptografi yang digunakan adalah AES (*Advanced Encryption Standards*). AES sendiri merupakan algoritma kriptografi yang didesain oleh Vincent Rijmen dan John Daemen asal (Pradana, 2011). Pada tahun 2000 algoritma Rijndael terpilih sebagai algoritma kriptografi yang selain aman juga efisien dalam implementasinya (Surian, 2006).

Salah satu metode yang umum digunakan dalam steganografi adalah metode *Least Significant Bit* (LSB). Metode ini banyak digunakan karena tidak terlalu kompleks dan pesan yang disembunyikan cukup aman (Nur, 2010). Selain itu, LSB merupakan salah satu metode steganografi yang paling sederhana (Prihanto, 2010). Berdasarkan pada hal-hal tersebut, maka pada penelitian ini menggunakan metode Least Significant Bit.

Format citra digital yang dipilih untuk menampung pesan pada penelitian ini adalah format PNG. Karena format PNG ini telah didukung oleh hampir seluruh telepon genggam (Soplanit dan Bandaria, 2007) dan memiliki ukuran yang tidak terlalu besar. Selain itu, fomat PNG dipilih karena teknik kompresi yang

digunakan pada PNG merupakan teknik kompresi yang *lossless* (Boutell, 1997). Hal ini berarti tidak ada nilai bit yang berubah pada saat kompresi dan dekompresi sehingga kemungkinan hilang atau rusaknya pesan rahasia tidak dapat terjadi (Priskilla, 2010).

Dengan fakta-fakta yang telah dipaparkan, maka penelitian yang dilakukan adalah berupa implementasi kriptografi dan steganografi dengan menggunakan media penampung pesan berformat PNG yang diintegrasikan ke dalam sistem operasi berbasis mobile Android.

II. TINJAUAN PUSTAKA

A. Steganografi

Kata steganografi (*steganography*) berasal dari bahasa Yunani steganos, yang artinya tersembunyi atau terselubung, dan graphia yang artinya menulis, sehingga arti steganografi adalah "menulis" (tulisan) terselubung" (Cvejic, 2004). Dengan steganografi, kita dapat menyisipkan pesan rahasia ke dalam media lain dan mengirimkannya tanpa ada yang menyadari keberadaan pesan tersebut (Krem, 2004). Penilaian sebuah algoritma steganografi yang baikdapat dinilai dari beberapa faktor yaitu (Alatas, 2009).

1. Impercepbility

Keberadaan pesan rahasia dalam media penampung tidak dapat dideteksi oleh inderawi. Misalnya, jika *covertext* berupa citra digital, maka penyisipan pesan membuat citra stegotext sukar dibedakan oleh mata dengan *covertext*-nya.

2. Fidelity

Mutu media penampung tidak berubah banyak akibat penyisipan. Perubahan itu tidak dapat dipersepsi oleh inderawi. Misalnya, jika *covertext* berupa citra, maka penyisipan pesan dapat membuat citra stegotext sukar dibedakan. Jika *covertext* berupa audio, maka audio stegotext tidak rusak dan indera telinga tidak dapat mendeteksi perubahan pada *filestegotext*-nya.

3. Recovery

Pesan yang disembunyikan harus dapat diungkapkan kembali. Karena tujuan steganografi adalah data hidding, maka sewaktu-waktu pesan rahasia di dalam *stegotext* dapat diambil kembali untuk digunakan lebih lanjut.

B. Least Significant Bit (LSB)

Metode steganografi yang paling umum pada tipe berkas citra adalah LSB (*Least Significant Bit*). Metode ini menyembunyikan data dengan mengganti bit-bit data yang paling tidak berarti di dalam *cover* dengan bit-bit data rahasia. Pada susunan bit di dalam sebuah *byte* (1 byte = 8 bit), ada bit yang paling berarti *Most Significant Bit* (MSB) dan bit yang paling kurang berarti *Least Significant Bit* (LSB). Bit yang cocok untuk diganti adalah LSB, sebab perubahan tersebut hanya mengubah nilai *byte* satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya. Misalkan pada *cover* citra, *byte* tersebut menyatakan warna merah, maka perubahan satu bit LSB tidak mengubah warna merah tersebut secara berarti, apalagi mata manusia tidak dapat membedakan perubahan kecil (Budiman, 2009).

C. Kriptografi

Kriptografi berasal dari dua kata Yunani, yaitu *Crypto* yang berarti rahasia dan *Grapho* yang berarti menulis. Menurut Bruce Scheiner, Secara umum kriptografi dapat diartikan sebagai ilmu dan seni penyandian yang bertujuan untuk menjaga keamananan dan kerahaisaan suatu data (Wahyudi, 2008).

Kriptografi memiliki dua bagian yang penting, yaitu enkripsi dan dekripsi. Enkripsi adalah sebuah proses penyandian pesan asli menjadi pesan yang tidak dapat diartikan seperti aslinya. Dekripsi sendiri berarti merubah pesan yang sudah disandikan menjadi I pesan aslinya. Pesan asli biasanya disebut plaintext, sedangkan pesan yang sudah disandikan disebut ciphertext (Sofwan, Budi P, & Susanto, 2006).

AES (*Advanced Encryption Standard*) – Rijndael merupakan algoritma kriptografi bernama Rijndael didesain oleh Vincent Rijmen dan John Daemen asal Belgia. Algoritma Rijndael inilah yang kemudian dikenal dengan AES (Advanced Encryption Standards) yang diadopsi menjadi standard algoritma kriptografi (Pradana, 2011). Rijndael mendukung panjang kunci 128 bit sampai 256 bit, maka dikenal dengan AES-128, AES-192, dan AES-256 (Wahyudi, 2008).

D. Citra Digital

Secara harafiah, citra (*image*) adalah gambar pada bidang dua dimensi (dwimatra). Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dwimatra. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut.Pantulan cahaya ini ditangkap oleh alat-alat optik, misalnya mata pada manusia, kamer. Pemindai (*scanner*), dan sebagainya, sehingga bayangan objek yang disebut citra tersebut terekam (Pratama, 2010).

PNG (*Portable Network Graphics*) adalah salah satu format penyimpanan citra yang menggunakan metode pemadatan yang tidak menghilangkan bagian dari citra tersebut (Inggris *lossless compression*). Untuk keperluan pengolahan citra, meskipun format PNG bisa dijadikan alternatif selama proses pengolahan citra, karena format ini selain tidak menghilangkan bagian dari citra yang sedang diolah (sehingga penyimpanan berulang ulang dari citra tidak akan menurunkan kualitas citra) PNG (Format berkas grafik yang didukung oleh beberapa web browser. PNG mendukung transparansi gambar seperti GIF, berkas PNG bebas paten dan merupakan gambar bitmap yang terkompresi (Saputra, 2011).

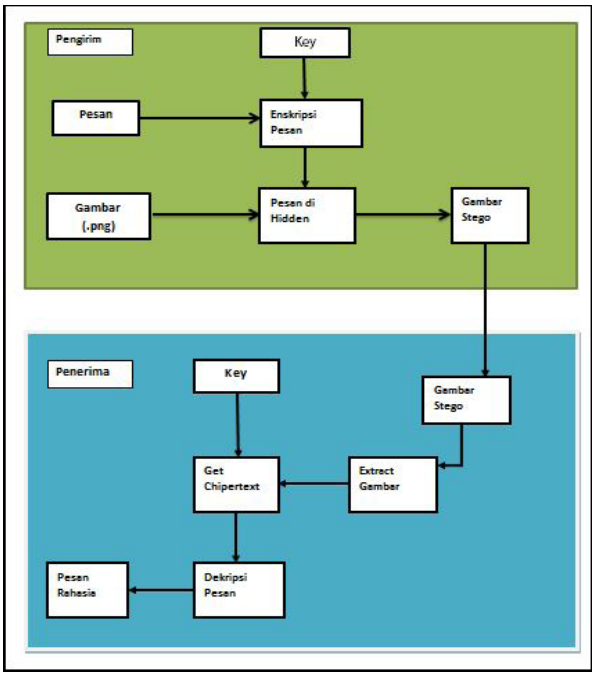
III. ANALISIS DAN PERANCANGAN

A. Metode Penelitian

Penelitian yang dilakukan melalui beberapa tahap. Tahap-tahap tersebut adalah studi literatur, perancangan aplikasi, pembuatan aplikasi, testing (uji coba), dan penulisan laporan..

B. Design Sistem

Perancangan aplikasi ini menggunakan metode steganografi, dimana dalam proses steganografi terdapat dua proses. Proses pertama adalah menyembunyikan pesan ke dalam media penampung pesan (*encode*). Dimana pesan yang disembunyikan ke dalam media dienkripsi terlebih dahulu. Proses kedua adalah pendeteksian pesan rahasia dari media penampung pesan (*decode*). Pada penelitian ini proses tersebut dipaparkan sebagai berikut :



Gambar 1. Rancangan aplikasi Steganografi

Berikut ini adalah input dan output dari aplikasi yang dirancang.

1. *Input*

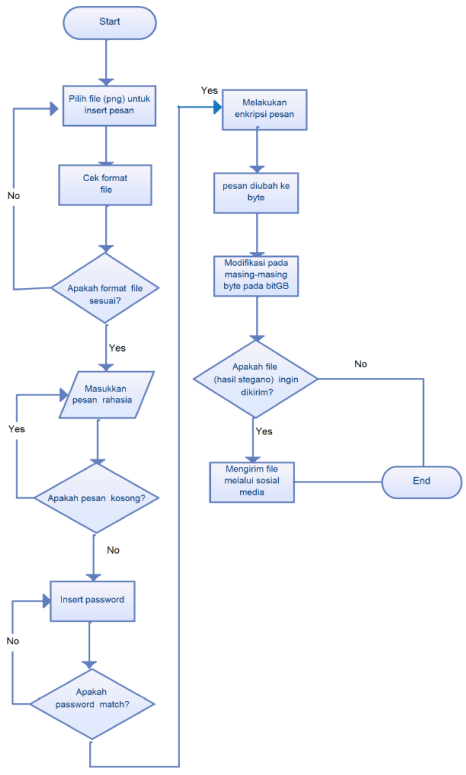
a. *Path* dari media penampung pesan berada yang dipilih oleh *user*, dimana media penampung pesan tersebut berupa *file* berformat PNG.
b. Pesan yang dimasukkan oleh *user* dalam bentuk teks (string).
c. *Password* yang berguna untuk enkripsi dan dekripsi pesan yang dimasukkan oleh *user*.

2. *Output*

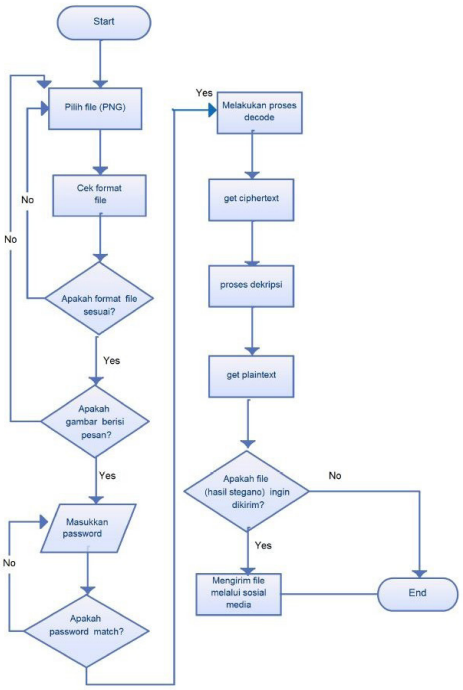
a. Pesan yang tersembunyi di dalam *file* gambar PNG dari proses *decode*.bentuk teks (string).
b. File PNG yang berisi pesan.

C. Flowchart

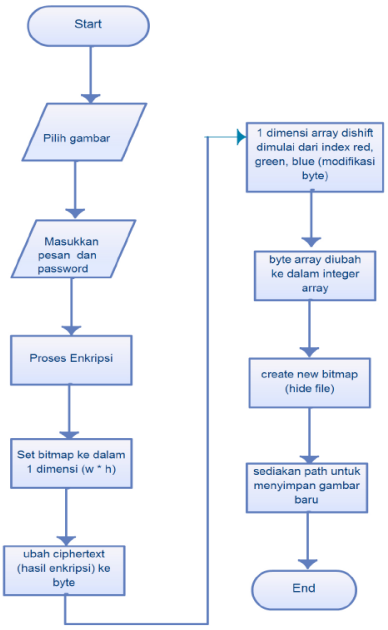
Berikut adalah gambaran dari *flowchart* sistem dari aplikasi yang mana terdiri dari *flowchart* sistem *hide* dan *extract*.



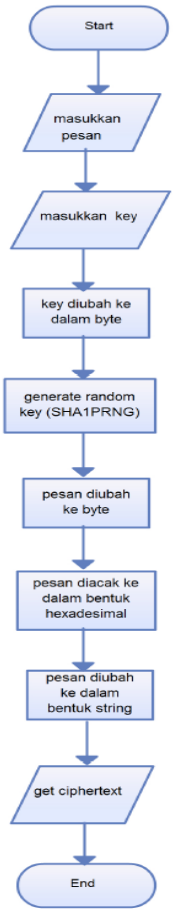
Gambar 2. Flowchart Sistem – Hide



Gambar 3. Flowchart Sistem – Extract

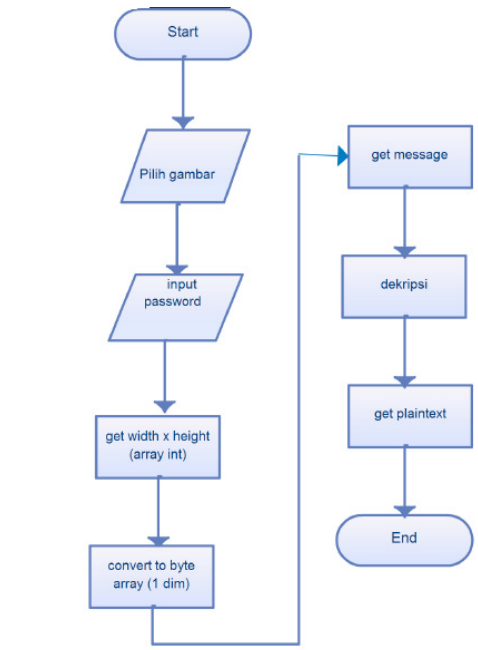


Gambar 5. Flowchart Encode



Gambar 4. Flowchart Enkripsi

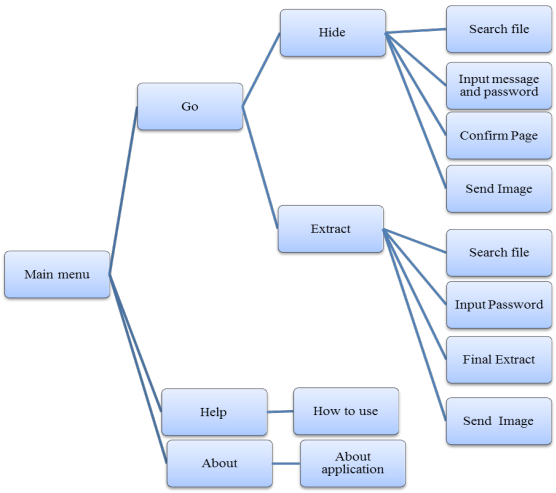
Gambar 6. Flowchart Dekripsi



Gambar 7. Flowchart Decode

D. Hierarki Menu

Berikut ini adalah top down design dari aplikasi steganografi.



Gambar 8. Top down design aplikasi

IV. IMPLEMENTASI SISTEM

A. Tampilan Implementasi

Berikut adalah rancangan antarmuka berupa *form-form* yang dibangun untuk mempermudah *user* berinteraksi dengan sistem. *Form-form* dikelompokan berdasarkan proses dari steganografi yaitu *hide message* dan *extract message*. Akan tetapi sebelum proses *hide message* dan *extract message*, terdapat beberapa tampilan awal seperti, Menu Utama, Menu Go, Help, dan About.

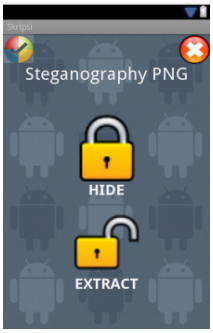
A.1. Tampilan Menu Utama

Halaman ini merupakan menu utama pada aplikasi, dimana terdapat 3 menu utama diantaranya Go, About, dan Help.

Gambar 9. Menu Utama

A.2. Menu Go

Pada tampilan menu Go, terdapat dua pilihan menu utama dari steganografi yaitu Hide dan Extract.



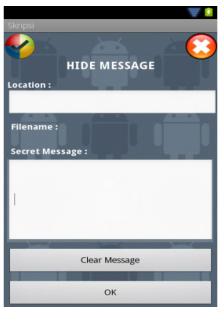
Gambar 10. Menu Go

A.3. Proses Hide Message

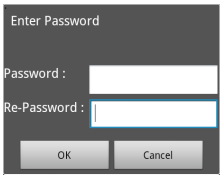
Tampilan menu Hide ditampilkan melalui Main Menu > Go > Hide. Pada tampilan Hide ini *user* diminta untuk memilih *file* PNG untuk melakukan *insert message*. Pada tampilan ini terdapat button OK dan Cancel. Jika *user* menekan button OK maka *user* akan dibawa ke tampilan layar selanjutnya. Jika user menekan button Cancel maka *user* dapat mengganti *file* lain.

Gambar 10. Menu Hide

Pada halaman hide message, *user* diminta untuk memasukkan pesan yang ingin disembunyikan. Jika user tekan button Clear Message, maka pesan yang *user* inputkan akan terhapus semua. Setelah selesai memasukkan pesan tekan button OK untuk melanjutkan proses *hidden message* kemudian akan ditampilkan *dialog password* seperti pada Gambar 12.

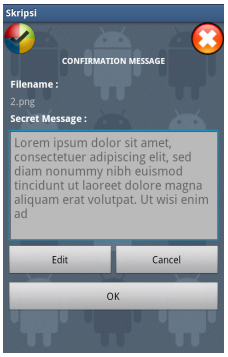


Gambar 11. Hide Message



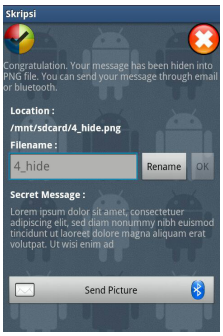
Gambar 12. Dialog Password

Setelah *password* yang dimasukkan telah sesuai kriteria maka proses dilanjutkan ke tampilan Confirmation Message. Pada tampilan ini *user* akan diperlihatkan pesan yang sudah dimasukkan sebelumnya dan *user* dapat ubah pesan dengan cara menekan button Edit.



Gambar 13. Confirmation Message

Setelah button OK ditekan, proses yang sistem lakukan adalah berupa enkripsi pesan dan *hidden* pesan. Setelah proses enkripsi dan *hide* pesan selesai maka user akan diberitahu bahwa pesan sudah selesai di *hidden* dan ditampilkan halaman seperti Gambar 14.

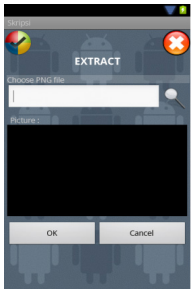


Gambar 14. Final Hide

A.4. Proses Extract Message

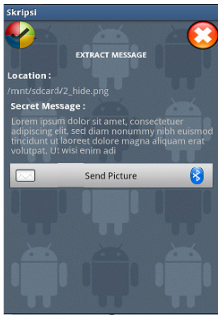
Berikut adalah tampilan-tampilan dari salah satu proses utama aplikasi yaitu Extract. Tampilan menu Extract ditampilkan melalui Menu Utama > Go > Extract. Pada tampilan Extrcat ini *user* diminta untuk memilih *file* PNG untuk melakukan *extract* pesan yaitu mengungkapkan kembali pesan yang berada dalam *file*.

User memilih *file* yang sudah berisi pesan. Kemudian tekan button OK maka akan ditampilkan *dialog* untuk *insert password*. *Password* yang diinputkan harus sesuai dengan *password* pada saat *user* melakukan *hide* pesan.



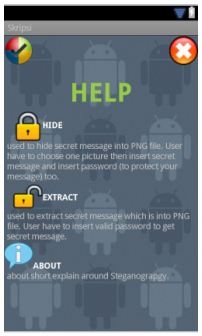
Gambar 15. Menu Extract

Setelah itu, maka sistem melakukan proses dekripsi pesan kemudian melakukan *decode* pesan. Lalu ditampilkan tampilan seperti di bawah ini yaitu berupa pesan.



Gambar 16. Final Extract

A.5. Menu Help



Gambar 17. Menu Help

A.6. Menu About

Gambar 18. Menu About

B. Pengujian Sistem

B.1. Data Pengujian

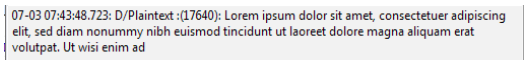
Uji coba dilakukan dengan menggunakan media penampung pesan berupa *file* PNG *true color* dengan ukuran ukuran pixel 277 x 277 (*size* : 4.77KB). Sedangkan pesan yang disisipkan menggunakan 160 karakter, 480 karakter, 1000 karakter, dan 6500 karakter.

B.2. Proses Pengujian

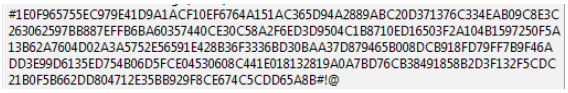
Masuk ke menu Hide, proses *generate key* secara *random* sesuai tipe enkripsinya yaitu 128 bit. Untuk mendapatkan bilangan *random*, digunakan SHA1PRNG yang terdapat pada *class* java.security. SecureRandom.Class.

Setelah melakukan *generate key* dengan 128 bit dilanjutkan dengan proses enkripsi. Pada proses enkripsi, *plaintext* diubah ke dalam bentuk byte dengan menggunakan *key* yang telah di-*generate* dilanjutkan dengan pemanggilan *cipher instance* berupa AES dan menggunakan mode ENCRYPT_MODE.

Hasil dari *plaintext* berupa *byte* diubah ke dalam bentuk bilangan hexadesimal dan kemudian hasil dari bilangan hexadesimal dikonversi ke dalam bentuk string. Hasil dari konversi dari hexadesimal ke string menghasilkan string dengan pesan acak berupa hexadesimal. Hasil tersebut yang disebut dengan *ciphertext*. Berikut gambaran *plaintext* (160 karakter) dan *ciphertext*.



Gambar 20. *Plaintext* (pesan yang disisipi)



Gambar 21. *Ciphertext* (hasil enkripsi)

Setelah proses enkripsi selesai, secara langsung proses *encode* pesan tersebut dilakukan. Pesan yang di-*encode* adalah *ciphertext*. Proses *encode* diawali dengan pengambilan pixel pada gambar yang dijadikan integer array satu dimensi. Kemudian dikonversi menjadi *byte* array. Sedangkan *ciphertext* yang berbentuk string pun dikonversi ke dalam byte ditujukan untuk melakukan penyisipan pesan pada gambar. Hal tersebut dilakukan untuk melakukan modifikasi pada masing-masing *byte* Red, Green, dan Blue. Dimana bit terkahir pada masing-masing RGB akan disisipkan pesan dengan operasi *shift*.

Seperti halnya *encode* yaitu mengambil ukuran pixel pada gambar dan dijadikan ke dalam array integer satu dimensi serta diubah ke dalam *byte* array. Dan masing-masing *byte* digeser untuk mengambil pesan yang terdapat pada gambar tersebut. Setelah isi pesan didapatkan, maka dilanjutkan dengan dengan proses dekripsi. Pesan yang didapatkan masih berupa pesan acak (hexadesimal).

Proses dekripsi diawali dengan melakukan konversi pesan dan *key* ke dalam byte. Kemudian melakukan

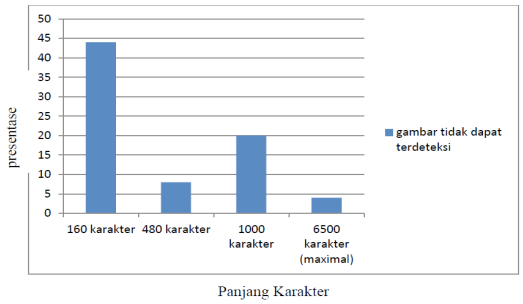
generate key untk mendapatkan *plaintext*. Kemudian pesan diubah kedalam bentuk string kembali agar pesan dapat dibaca.

B.3. Hasil Pengujian

Pengujian dilakukan dengan melakukan wawancara kepada 25 responden. Wawancara dilakukan dengan menunjukkan *file* gambar asli dengan empat *file* gambar yang sudah berisi pesan.

Berdasarkan hasil pengujian tersebut, terdapat 76% (responden) responden yang tidak dapat membedakan antara mana file asli dan mana file yang berisi pesan-pesan tersebut. Sedangkan 24% (6 responden) responden mengetahui dengan benar file asli yang tidak berisi pesan. Dengan demikian salah satu kriteria steganografi yaitu *Impercebility* dapat terpenuhi. Dimana koresponden tidak dapat mendeteksi tentang keberadaan pesan yang terdapat pada gambar, karena tidak melihat adanya perbedaan pada setiap gambar tersebut.

Dibawah ini adalah grafik presentasi dari hasil perbandingan gambar.



Gambar 22. Grafik perbandingan pesan pada gambar

Kriteria kedua dari steganografi adalah *Fidelity*. Ternyata kriteria tidak terpenuhi dengan baik dalam metode LSB karena gambar yang berisi pesan mengalami perubahan fisik yaitu pembengkakan ukuran gambar. Seperti dapat dilihat pada tabel berikut ini :

Tabel 1. Tabel uji coba penyisipan pesan

File (.png)	Ukuran (KB)	Panjang Pesan	Ukuran akhir (KB)
Android1.png	4.74 KB (277 x277)	160 karakter	11.6 KB
		480 karakter	12.8 KB
		1000 karakter	14.6 KB
		6500 karakter	29 KB

Dari tabel hasil uji coba diatas dapat dilihat bahwa panjang pesan yang digunakan uji coba berbeda dengan merubah ukuran *file* pada gambar yang mengakibatkan bertambahnya *size* pada media penampung.

Berdasarkan hasil dari uji coba di atas, dapat maka dapat diperoleh presentase dari pembengkakkan sizepada gambar dengan menggunakan persamaan :

1. Gambar dengan 160 karakter

$$\frac{11.6 - 4.74}{11.6} \times 100\% = 59.1\ \%$$

2. Gambar dengan 480 karakter

$$\frac{12.8 - 4.74}{12.8} \times 100\% = 62.9\ \%$$

3. Gambar dengan 1000 karakter

$$\frac{14.6 - 4.74}{14.6} \times 100\% = 67.5\ \%$$

4. Gambar dengan 6500 karakter

$$\frac{29.4 - 4.74}{29.4} \times 100\% = 83.8\%$$

Kriteria ketiga dari steganografi adalah *Recovery*. Kriteria tersebut terpenuhi yaitu dengan menyisipkan pesan dengan jumlah karakter yang berbeda-beda yaitu dengan 160 karakter, 480 karakter, 1000 karakter, dan 6500 karakter. Kemudian hasil dari *extract* gambar sesuai dengan penyisipan pesan yang telah dilakukan sebelumnya.

V. SIMPULAN DAN SARAN

A. Simpulan

Berdasarkan hasil dari penelitian yang telah dilakukan dapat disimpulkan bahwa aplikasi steganografi berhasil diimplementasikan pada *mobile phone* Android. Aplikasi ini dapat melakukan penyisipan pesan dari 160 karakter, 480 karakter, 1000 karakter hingga 6.500 karakter sesuai dengan gambar yang telah dilakukan uji coba.

Berdasarkan kriteria steganografi bahwa aspek *impercebility* dan *recovery* dapat terpenuhi dengan baik. Pada aspek *impercepbility* gambar yang berisi pesan secara kasat mata tidak dapat dibedakan dengan pesan asli karena hanya terjadi sedikit perubahan warna pada gambar yang disisipkan pesan. Hal ini memunjukkan bahwa kualitas gambar dari file PNG yang telah disisipkan pesan memiliki kualitas yang baik.

Aspek *recovery* dapat dibuktikan dengan dapat diungkapkan kembali isi pesan yang terdapat dalam citra digital PNG. Salah satu kriteria steganografi yang tidak terpenuhi adalah *fidelity* karena pada hasil uji

coba, terjadi pembengkakan *size* pada gambar yang berisi pesan.

B. Saran

Untuk pengembangan lebih lanjut, beberapa saran yang diberikan adalah melakukan kompresi pada pesan agar *file* pada media penampung tidak mengalami pembengkakan *size* yang terlalu banyak, sehingga pesan yang disisipkan dapat lebih banyak lagi. Aplikasi dapat dikembangkan lebih lanjut dengan menggunakan media penampung selain format gambar PNG. Selain itu, aplikasi sebaiknya dibandingkan dengan metode steganografi yang lain untuk mendapatkan hasil terbaik dalam penyisipan pesan. Aplikasi dapat diimplementasikan pada sistem operasi atau aplikasi mobile lain seperti, Blackberry.

Daftar Pustaka

[1] Alatas, Putri. 2009. "Implementasi Teknik Steganografi D Bit engan Metode LSB Pada Citra Digital".

[2] Budiman, Asep.2009. "Aplikasi Steganography Pada Video Dengan Metode Least Significant Bit (LSB)".

[3] Firmansyah, Rizqi. 2011. "Implementasi Kriptografi dan Steganografi Pada Media Gambar Dengan Menggunakan Metode DES dan Region Embed Data Density".

[4] Ginting, Priskilla BR. 2010. "Kajian Steganografi dengan Metode Bit Plane Complexity Segmentation (BPCS) Pada Dokumen Citra Terkompresi".

[5] Pradana, Ridhky Oktavian. 2011. "Analisis Perbandingan Algoritma Rijndael dan Algoritma Twofish Pada Proses Pengiriman Data Teks Menggunakan Jaringan LAN (Local Are Network)".

[6] Pratama, Andika. 2011. "Eksplorasi Penerapan Steganografi Dengan Eksploitasi Spesifikasi Format Media Container Popular".

[7] Prihanto, Agus. 2010. "Peningkatan Kapasitas Informnasi Tersembunyi Pada Image Steganografi Menggunakan Teknik Hybrid".

[8] Saputra, Hasbian. 2011. "Implementasi Algoritma Steganografi Embedding Dengan Metode Least Significant Bit (LSB) Insertion Dan Huffman Coding Pada Pengiriman Pesan Menggunakan Media MMS Berbasis J2ME".

[9] Sofwan, A., Budi P, A., & Susanto, T. 2006. "Aplikasi Kriptografi dengan Algoritma Message Digest 5(MD5)".

[10] Soplanit, Susany & Bandaria, Constatine. 2007. "Steganografi Dengan Chaotic Least Significant Bit Encoding Pada Telepon Genggam".

[11] Surian, Didi. 2006. "Algoritma Kriptografi AES Rijndael". TESLA Vol. 8 No. 2, 97 – 101.

[12] Tumanggor, Seti Fauziah. 2009. "Studi Enkripsi Dan Dekripsi File Dengan Menggunakan Algoritma Twofish".

[13] Wahyudi, Kunjung. 2008. "Aplikasi Steganografi Untuk Pertukaran Pesan Dengan Menggunakan Teknik Steganografi Dan Algoritma AES.