Kumpulan jurnaL Ilmu Komputer (KLIK)
Volume 06, No.01 Februari 2019
ISSN: 2406-7857

STEGANOGRAFI UNTUK PESAN TERENKRIPSI MENGGUNAKAN ALGORITMA KRIPTOGRAFI RSA-CRT DI ANDROID

Rudy Herteno[1], Wahyu Ramadansyah[2], Oni Soesanto[3], Radityo Adi Nugroho[4], Ahmad Rusadi Arrahimi[5]

[1,2,4,5]Prodi Ilmu Komputer FMIPA ULM
[3]Prodi Statistika FMIPA ULM
Jl. A. Yani Km 36 Banjarbaru, Kalimantan Selatan
[1]Email : rudy.herteno@gmail.com

Abstract

In computer science, one of the methods commonly used as a message security algorithm is LSB (Least Significant Bit) steganography. Steganography LSB is generally used without a message security system that avoids the influence of messages from the effort to eliminate the inserted message so that it needs to be developed with a combination of additional methods namely RSA-CRT algorithm. The implementation of the LSB method with the RSA-CRT algorithm can improve the security of the message to be inserted. The secret message is encrypted first using the RSA-CRT algorithm and then inserted in the image. The purpose of this study was to determine the feasibility of RSA-CRT cryptographic algorithms and LSB steganography implemented on android. Testing is done by measuring the processing time on different smartphones and different android versions as well. The result is the Android version and smartphone specifications do not affect the speed of the process of the message security application using RSA-CRT cryptographic algorithms and LSB steganography on Android.

***Keywords**: Cryptography, Steganography, RSA-CRT, LSB, Android*.

Abstrak

Dalam ilmu komputer salah satu metode yang umumnya digunakan sebagai algoritma pengamanan pesan adalah steganografi LSB (Least Significant Bit). Steganografi LSB umumnya digunakan tanpa adanya suatu sistem keamanan pesan yang menghindari terpengaruh nya pesan dari upaya menghilangkan pesan yang disisipkan sehingga perlu dikembangkan dengan kombinasi metode tambahan yaitu algoritma RSA-CRT. Implementasi metode LSB dengan algoritma RSA-CRT dapat meningkatkan keamanan pesan yang akan disisipkan. Pesan rahasia dienkripsi terlebih dahulu menggunakan algoritma RSA-CRT lalu disisipkan dalam gambar. Tujuan dari penelitian ini adalah mengetahui kelayakan algoritma kriptografi RSA-CRT dan steganografi LSB di implementasikan di android. Pengujian dilakukan dengan cara mengukur waktu proses di smartphone yang berbeda dan versi android yang berbeda juga. Hasilnya versi android dan spesifikasi smartphone tidak mempengaruhi waktu kecepatan proses aplikasi pengamanan pesan menggunakan algoritma kriptografi RSA-CRT dan steganografi LSB di android.

***Kata kunci**:* Kriptografi, Steganografi, RSA-CRT, LSB, Android.

1. PENDAHULUAN

Penggunaan teknologi telepon genggam (*handphone*) saat ini sebagai media utama dalam berkomunikasi dapat menggantikan fungsi komputer dalam proses komunikasi, *handphone* memiliki cara-cara yang digunakan dalam berkomunikasi yang dapat digunakan antara lain, SMS, MMS, *chatting*, dan *e-mail*. Seiring dengan perkembangan zaman muncul beberapa inovasi terbaru pada telepon genggam diantaranya sistem operasi yang mencakup segala macam kebutuhan seperti komputer, yang bisa disebut dengan *Android*. *Android* merupakan sistem operasi untuk telepon genggam yang mencakup sistem operasi, aplikasi dan menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi.[1]

Selama ini aplikasi yang digunakan untuk mengirim dan menerima pesan di android belum bisa menjamin privasi diantara pengirim dan penerima pesan, karena pesan yang dikirim merupakan pesan yang langsung dapat dibaca tanpa ada pengamanan tambahan. Berbagai aplikasi perpesanan yang ada di android saat ini belum tentu aman untuk digunakan karena tidak ada standar tertentu dalam pengamanan pesan yang digunakan oleh aplikasi-aplikasi tersebut, Untuk itulah akan dibuat suatu aplikasi pengamanan pesan berbasis Android yang berfungsi untuk enkripsi (*encryption*) dan dekripsi (*decryption*) pesan yang telah dibuat tersebut.[2]

Sandro menjelaskan bahwa untuk melakukan pengamanan pesan dilakukan proses enkripsi pesan menggunakan metode kriptografi RSA-CRT dan metode steganografi LSB. Algoritma RSA-CRT adalah RSA yang di modifikasi dengan Teorema Sisa Cina atau CRT (*Chinese Remainder Theorem*). Sedangkan steganografi merupakan metode yang digunakan untuk menyembunyikan suatu pesan atau data rahasia di dalam suatu media penampungnya sehingga orang lain tidak menyadari adanya pesan di dalam media tersebut.[3]

Berdasarkan uraian diatas maka di perlukan algoritma kriptografi RSA-CRT dan algoritma steganografi LSB dalam pengamanan pesan berbasis android. Tujuan nya agar terbentuknya aplikasi untuk pengamanan pesan secara aman dan fleksibel di android.

2. METODOLOGI PENELITIAN

2.1. Kriptografi

Menurut Ashari Arief dan Ragil Saputra Kriptografi adalah suatu ilmu yang mempelajari teknik matematika yang berhubungan dengan keamanan informasi seperti kerahasiaan data, integritas data, otentikasi entitas, dan otentikasi asal data. Kriptografi bertujuan agar informasi yang bersifat rahasia dan dikirim melalui suatu jaringan, seperti LAN atau internet, tidak dapat diketahui dan dimanfaatkan oleh orang lain selain si pengirim dan penerima atau pihak lain yang tidak berkepentingan.[4]

RSA merupakan algoritma kriptografi kunci publik atau sering disebut kunci asimetrik (kunci enkripsi dan kunci dekripsi berbeda) sehingga tidak membutuhkan saluran yang aman untuk distribusi kunci. RSA ditemukan oleh tiga

peneliti dari MIT (*Massachussets Institute of Technology*), yaitu Ronald Linn Rivest, Adi Shamir, dan Len Adleman pada tahun 1977. Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima.

Andri Risky Dinata mengatakan bahwa algoritma kriptografi terbagi menjadi dua, yaitu algoritma simetris (*symmetric algoritm)* dan algoritma asimetris (*asymmetric algoritm)*.

a. Algoritma Simetris
   Algoritma simetris pada kriptografi merupakan proses enkripsi dan deskripsi nya menggunakan satu kunci. Algoritma simetris dapat diasumsikan bahwa si penerima dan pengririm pesan telah terlebih dahulu berbagi kunci sebelum pesan dikirimkan., keamanan dari sistem ini tergantung pada kerahasiaan kuncinya, jika kunci tersebut diketahui orang lain selain pengirim dan penerima maka orang tersebut dapat melakukan proses enkripsi dan deskripsi pesan.[5]

b. Algoritma Asimetris
   Algoritma asimetris pada kriptografi mengacu pada proses enkripsi dan deskripsi pada algoritma asimetris menggunakan *key* yang berbeda biasanya disebut dengan istilah kunci enkripsi *(public key)* dan kunci dekripsi *(private key)*. Kunci untuk proses enkripse tidak bersifat rahasia, tetapi kunci untuk proses deskripsi bersifat rahasia. Sehingga apabila ada orang lain yang mengetahui kunci publik saja maka dia tidak bisa melakukan proses deskripsi pesan karena memakai kunci rahasia yang berbeda dari kunci publik.[5]

Teorema Sisa Cina atau CRT (*Chinese Remainder Theorem*) merupakan suatu algoritma untuk mengurangi perhitungan aritmatika modular dengan modulus besar untuk perhitungan yang sama untuk masing-masing faktor dari modulus. CRT dapat memperpendek ukuran bit eksponen dekripsi d (merupakan kunci publik RSA atau RSA-CRT) caranya dengan menyamarkan d dalam sebuah sistem yang kongruen supaya dapat meningkatkan kecepatan waktu dekripsi pesan serta dapat digunakan bersama algoritma RSA yang disebut RSA-CRT[5].

Umumnya algoritma RSA-CRT tidak ada bedanya dengan RSA biasa tetapi perbedaan nya adalah menggunakan metode sisa cina atau CRT digunakan untuk mengurangi ukuran dari bit eksponen deksipsi d caranya dengan menyembunyikan d didalam sebuah sistem yang kongruen sehingga dapat mempercepat waktu dekripsi pesan. Berikut algoritma pembangkit kunci RSA-CRT:

a. Bangkitkan bilangan prima besar p dan q
b. Hitung nilai modulus n = $p \times q$ ...(1)
c. Hitung fungsi Euler n menggunakan f(n) = (p-1) x (q-1) ...(2)
d. Pilih nilai integer e secara acak sebagai kunci publik. Dengan syarat memenuhi Faktor Persekutuan Terbesar (FPB) (e, f(n)) = 1,1<e<f(n) ...(3)
e. Hitung kunci privat d sehingga d x e = 1(mod f(n)) ...(4)
f. Hitung nilai dP = d mod (p-1) ...(5)

g. Hitung nilai dQ = d mod (q-1) ...(6)
h. Hitung nilai qInv = q^{-1} mod p ...(7)

Kunci publik RSA-CRT sama dengan sistem RSA yaitu (e, n) sehingga algoritma enkripsi tidak mengalami perubahan yaitu dengan menggunakan fungsi eksponensial modular yaitu : $C = M^e \bmod n$ ...(8)

Proses dekripsi menggunakan algoritma RSA-CRT dilakukan dengan menggunakan rumus :

a. $m_1 = C^{dP} \bmod p$ ... (9)
b. $m_2 = C^{dQ} \bmod q$ ...(10)
c. $h = qInv(m_1 - m_2) \bmod p$ ...(11)
d. $M = m_2 + h.q$ ...(12)

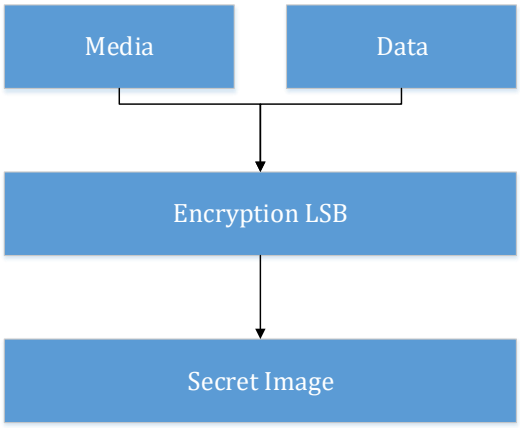
2.2. Steganografi

Steganografi adalah metode atau algoritma untuk mengenkripsi pesan tersembunyi dengan cara-cara lain jadi selain pengirim dan penerima pesan tidak ada yang dapat mengetahui kalau ada pesan rahasia yang tersimpan. Sebaliknya, kriptografi adalah ilmu tentang menyembunyikan maksud dari suatu pesan agar tidak diketahui oleh orang yang tidak bertanggung jawab, akan tetapi tidak menyamarkan keberadaan pesan terenkripsi. Makna dari steganografi berasal dari bahasa Yunani yaitu *steganos*, yang artinya terselubung atau tersembunyi, dan *graphein*, yaitu menulis. Pada saat ini istilah steganografi umumnya digunakan untuk menyisipkan data digital dalam *file* komputer.[6]

Steganografi sekarang ini digunakan sebagai metode untuk saling berkomunikasi dan berfungsi sebagai metode dalam mengenkripsi pesan rahasia di dalam file komputer tanpa adanya menunjukkan perubahan yang signifikan yang dapat dilihat secara kasat mata dan struktur dari berkas semula. Teknik steganografi juga termasuk dalam *invisible ink*, *microdots*, tanda tangan elektronik, jalan rahasia dan juga komunikasi spektrum lebar. Steganografi bertujuan untuk menyembunyikan keberadaan dari suatu pesan atau informasi rahasia. Dalam penggunaan nya, steganografi umumnya merupakan pesan yang disembunyikan dengan membuat perubahan yang tidak signifikan terhadap data digital yang isinya tidak akan menarik perhatian dari pihak yang tidak bertanggung jawab, contohnya, ada pesan masuk yang berupa gambar yang terlihat biasa saja. Proses perubahan steganografi tergantung pada kunci enkripsi dan juga tergantung isi pesan yang ingin disembunyikan. Si penerima pesan atau penerima menerima gambar dapat menyimpulkan informasi rahasia yang tersimpan dengan cara mengubah *key* yang sesuai ke dalam metode yang digunakan.[6]

Sebuah pesan yang mempunyai stegano, biasanya pertama kali di enkripsi atau disisipkan sebuah pesan, yang menghasilkan *ciphertext*. Ciphertext merupakan hasil dari penyisipan pesan menggunakan algoritma steganografi.[6]

Untuk proses steganografi yang terdapat pada penelitian adalah proses penyisipan pesan yang berupa *text* kedalam media gambar dengan format *jpg, png,* dan *jpeg*. Untuk cara kerja steganografi akan ditampilkan pada gambar sebagai berikut :



Gambar 1. Cara kerja Steganografi

3. HASIL DAN PEMBAHASAN

3.1 HASIL

Hasil dari implementasi algoritma steganografi LSB dan kriptografi RSA-CRT sebagai metode pengamanan pesan rahasia di android adalah sebagai berikut :

Gambar 2. Proses *generate key*

Gambar 3. Proses enkripsi pesan

Generate key berfungsi sebagai menu untuk membuat kunci yang akan digunakan dalam proses enkripsi pesan dan proses dekripsi pesan rahasia. Inputan yang ada pada menu *generate key* merupakan 2 bilangan prima acak.

Selanjutnya enkripsi pesan yang digunakan untuk mengenkripsi pesan rahasia menggunakan algoritma RSA-CRT. Inputan pada halaman enkripsi pesan berupa pesan teks yang ingin dienkripsi atau dirahasiakan makna nya. Keluaran yang dihasilkan berupa deretan bilangan ASCII yang merupakan hasil enkripsi algoritma kriptografi RSA-CRT.

Setelah pesan yang ingin dienkripsi berhasil diproses menggunakan algoritma kriptografi RSA-CRT maka proses selanjutnya adalah menyisipkan pesan rahasia tersebut kedalam media citra gambar dengan format *jpg*, *jpeg*, dan *png* menggunakan algoritma steganografi LSB. Pesan yang terenkripsi berupa bilangan ASCII tersebut akan disisipkan kedalam media gambar dengan mengubah bit

terakhir dari pixel gambar tersebut. Untuk tampilan hasil enkripsi kedalam gambar adalah sebagai berikut :

Gambar 4. Hasil enkripsi pada gambar *png*

Gambar 5. Hasil enkripsi pada gambar *jpg*

Gambar 6. Hasil enkripsi pada gambar *jpeg*

Selanjutnya setelah pesan di enkripsi dan dikirimkan kepada penerima maka penerima akan mendekripsi pesan, dekripsi pesan digunakan untuk mengeluarkan pesan rahasia yang terdapat dalam gambar tersebut sekaligus mendekripsi pesan terenkripsi didalam nya. Inputan yang ada pada menu dekripsi pesan berupa gambar yang diterima dari pengirim pesan, keluaran dari menu dekripsi pesan adalah isi pesan yang dikirimkan oleh pengirim pesan tersebut. Untuk tampilan halaman dekripsi pesan sebagai berikut:

Gambar 6. Isi pesan rahasia

3.2 PEMBAHASAN

Dalam penelitian ini menggunakan data *sample* waktu tentang proses-proses yang ada, diantaranya adalah *generate key,* enkripsi pesan, enkripsi kedalam gambar, dan dekripsi pesan. Tiap proses diambil 10 data *sample* yang akan digunakan untuk menghitung nilai rata-rata. Kemudian nilai rata-rata tersebut dijumlahkan untuk mendapatkan jumlah waktu rata-rata proses yang diperlukan untuk menjalankan aplikasi. Pengambilan *sample* dilakukan pada 3 *smartphone* android yang memiliki klasifikasi rendah, sedang, dan tinggi. Setiap *smartphone* diuji coba menggunakan 3 versi android yang saat ini banyak digunakan yaitu *lollipop, marshmallow,* dan *nougat* dan diukur waktu pemrosesan aplikasi stegano ini.

Untuk *smartphone* yang digunakan diklasifikasikan berdasarkan kapasitas RAM yang ada, *smartphone* dengan kategori rendah memiliki kapasitas RAM sebesar 2 GB (*gigabyte*), kategori sedang memiliki kapasitas RAM 3 GB (*gigabyte*), dan kategori tinggi memiliki kapasitas RAM 4 GB (*gigabyte*). Berikut ini daftar *smartphone* yang digunakan dalam pengujian ini:

Tabel 1. Daftar *smartphone* yang digunakan dalam pengujian

Nama *smartphone*	**Kapasitas RAM**
Asus zenfone live	2 GB
Xiaomi note 3 pro	3 GB
Xiaomi note 5	4 GB

Setelah itu hasil waktu yang didapat akan diuji dengan metode *Two-way Anova* dan diamati hasilnya untuk ditarik kesimpulan sesuai dengan syarat H_0, H_1, dan H_2. Selanjutnya pengujian yang dilakukan adalah proses enkripsi pesan dengan algoritma kriptografi RSA-CRT kemudian menyisipkan pesan kedalam gambar dengan algoritma steganografi LSB. Kemudian pesan tersebut dikirimkan ke penerima dan di dekripsi untuk mengetahui isi pesan yang tersembunyi didalam gambar tersebut.

Pesan yang digunakan untuk pengujian memiliki panjang 12 karakter huruf, Misalnya pesan tersebut adalah "*meet at park*". Dengan panjang pesan yang sama diukur kecepatan proses enkripsi di *smartphone* dengan versi android yang berbeda-beda. Gambar yang digunakan sebagai media penyisipan pesan adalah gambar dengan tipe jpg, jpeg, png. Setiap *smartphone* android melakukan percobaan menyisipkan pesan kedalam gambar dengan tipe jpg, jpeg, dan png.

Hasil pengujian pada *smartphone* yang memiliki spesifikasi rendah diwakili oleh *Asus zenfone live* yang memiliki kapasitas RAM 2 gb, hasil pengujian yang dilakukan adalah semua proses yang ada dalam aplikasi stegano yaitu proses *generate key*, enkripsi pesan, enkripsi pesan kedalam gambar, dan dekripsi pesan. Kemudian hasil dari semua proses tersebut dicari nilai rata-rata nya dan dijumlahkan maka di peroleh hasil sebagai berikut :

Tabel 2. Hasil pengujian pada *smartphone* 2 gb

Versi android	**Waktu**
android lollipop	5.3686 s
android marshmallow	5.6493 s
android nougat	4.4921 s

Hasil pengujian pada *smartphone* yang memiliki spesifikasi sedang diwakili oleh *Xiaomi note 3* yang memiliki kapasitas RAM 3 gb, hasil pengujian yang dilakukan adalah semua proses yang ada dalam aplikasi stegano yaitu proses *generate key*, enkripsi pesan, enkripsi pesan kedalam gambar, dan dekripsi pesan. Kemudian hasil dari semua proses tersebut dicari nilai rata-rata nya dan dijumlahkan maka di peroleh hasil sebagai berikut :

Tabel 3. Hasil pengujian pada *smartphone* 3 gb

Versi android	**Waktu**
android lollipop	3.5837 s
android marshmallow	4.0028 s
android nougat	3.109 s

Hasil pengujian pada *smartphone* yang memiliki spesifikasi tinggi diwakili oleh *Xiaomi note 5* yang memiliki kapasitas RAM 4 gb, hasil pengujian yang dilakukan adalah semua proses yang ada dalam aplikasi stegano yaitu proses *generate key*, enkripsi pesan, enkripsi pesan kedalam gambar, dan dekripsi pesan. Kemudian hasil dari semua proses tersebut dicari nilai rata-rata nya dan dijumlahkan maka di peroleh hasil sebagai berikut :

Tabel 4. Hasil pengujian pada *smartphone* 4 gb

Versi android	Waktu
android lollipop	2.368
android marshmallow	3.2624
android nougat	4.3771

Kemudian hasil dari pengujian waktu dengan menggunakan *smartphone* dengan klasifikasi rendah, sedang, dan tinggi menggunakan versi android lollipop, marshmallow, dan *nougat* diatas dijabarkan pada tabel dibawah ini :

Tabel 5. Hasil pengujian dengan jumlah waktu rata-rata

	Zenfone live	Xiaomi note 3	Xiaomi note 5
android lollipop	5.3686 s	3.5837 s	2.368 s
android marshmallow	5.6493 s	4.0028 s	3.2624 s
android nougat	4.4921 s	3.109 s	4.3771 s

Pada *smartphone* Asus zenfone live menggunakan android *lollipop* didapatkan total waktu rata-rata untuk memproses aplikasi sebesar 5.3686 *second.* Pada android *marshmallow* didapatkan total waktu rata-rata untuk memproses aplikasi sebesar 5.6493 *second*, dan pada android *nougat* didapatkan total waktu rata-rata untuk memproses aplikasi sebesar 4.4921 *second*.

Pada *smartphone* Xiaomi note 3 menggunakan android *lollipop* didapatkan total waktu rata-rata untuk memproses aplikasi sebesar 3.5837 *second*. Pada android *marshmallow* didapatkan total waktu rata-rata untuk memproses aplikasi sebesar 4.0028 *second,* dan pada android *nougat* didapatkan total waktu rata-rata untuk memproses aplikasi sebesar 3.109 *second*.

Pada *smartphone* Xiaomi note 5 menggunakan android *lollipop* didapatkan total waktu rata-rata untuk memproses aplikasi sebesar 2.368 *second*. pada android *marshmallow* didapatkan waktu total untuk memproses aplikasi sebesar 3.2624 *second,* dan pada android *nougat* didapatkan total waktu rata-rata untuk memproses aplikasi sebesar 4.3771 *second*.

Tabel 6. Hasil perhitungan Two Way Anova

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Android	0.42788486	2	0.21394	0.313799972	0.747150976	6.944272
Spesifikasi	5.992251167	2	2.996125	4.394565981	0.097822295	6.944272
Error	2.727118533	4	0.681779			
Total	**9.14725456**	**8**				

Hasil dari perhitungan menggunakan Two Way Anova dan hasil jumlah waktu rata-rata seperti pada tabel 2. Untuk versi android didapatkan nilai F = 0.313799972, nilai F crit = 6.944272. Untuk spesifikasi didapatkan nilai F = 4.394565981, nilai F crit = 6.944272. nilai P-value pada versi android didapatkan sebesar = 0.747150976, dan P-value pada spesifikasi didapatkan sebesar = 0.097822295. Karena nilai P-value > α = 0.05 maka H_0 ditolak. Karena nilai F < F crit maka H_1 ditolak. Dan karena nilai P-value > α = 0.05 dan nilai F < F crit maka H_2 ditolak. Berdasarkan hasil dari perhitungan Two Way Anova didapatkan bahwa spesifikasi *smartphone* dan versi android tidak mempengaruhi waktu proses aplikasi stegano.

4. SIMPULAN

Dari hasil penelitian dan pengamatan dari sistem yang telah dilakukan, dapat disimpulkan bahwa spesifikasi *smartphone* dan versi android tidak mempengaruhi waktu proses aplikasi sehingga aplikasi yang dibangun layak digunakan pada *smartphone* android.

DAFTAR PUSTAKA

[1] Prafanto, Anton.2016.**Penerapan Algoritma Blowfish Untuk Keamanan SMS pada Android**.Magister Teknik Elektro, Sekolah Tinggi Teknik Elektro dan Informatika,Instite Teknologi Bandung.

[2] Alvianto, Andi Riski & Darmaji.2015.**Pengamanan Pengiriman Pesan Via SMS Dengan Algoritma RSA Berbasis Android**.Jurusan Matematika,Fakultas MIPA,Institut Teknologi Sepuluh November, Surabaya.

[3] Sembiring, Sandro.2014.**Perancangan Aplikasi Steganografi Untuk Menyisipkan Pesan Teks Pada Gambar Dengan Metode End Of File**.Program Studi Teknik Informatika, STIMIK Budi Darma Medan.

[4] Arief, Ashari & Ragil Saputra.2016. **Implementasi Kriptografi Kunci Publik dengan Algoritma RSA-CRT pada Aplikasi Instant Messaging**. Jurusan Ilmu Komputer, Universitas Diponegoro, Semarang.

[5] Dinata, Andri Risky .2013.**Penggunaan Teorema Sisa Cina Pada Algoritma RSA.** Jurusan Matematika, Fakultas MIPA, Universitas Lambung Mangkurat.
[6] Utomo, Tri Prasetyo.2012. **Steganografi Gambar Dengan Metode Least Significant Bit Untuk Proteksi Komunikasi Pada Media Online**. Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, UIN Sunan Gunung Djati Bandung.