Penerapan Teknik Steganalysis Menggunakan Metode Chi Square Attack Pada Stego Image Berformat Jpeg Berbasis Android

Damayanti[1], Yudo Bismo Utomo[2], Halimahtus Mukminna[3]
[1,2]Teknik Elektro, Fakultas Teknik, Universitas Islam Kadiri Kediri
E-mail: *[1]damalus85@gmail.com, [2]yudobismo@uniska-kediri.ac.id, [3]halimahtusm@uniska-kediri.ac.id

Abstrak – Steganografi merupakan seni dalam menulis pesan rahasia untuk menyembunyikan informasi dengan cara meniyisipkan pesan pada suatu citra, sehingga tidak menutup kemungkinan steganografi dimanfaatkan untuk tujuan kriminal atau penyisipan *file* berbau pornografi. Sedangkan steganalysis merupakan teknik untuk mendeteksi keberadaan pesan yang tersembunyi pada *cover image*. Pada penelitian ini menggunakan teknik steganalysis dengan metode *chi-square attack* untuk mendeteksi keberadaan pesan secara statistik. Tujuan dari penelitian ini adalah mengembangkan teknik steganalysis dengan menggunakan metode *chi-square attack* untuk mendeteksi *stego-image* berformat JPEG dan mengetahui hasil analisa dari metode *chi-square attack* yang diterapkan pada aplikasi Android Studio. Metode yang digunakan pada penelitian ini adalah metode *waterfall*, yang terdiri dari tahap analisis kebutuhan, tahap desain, tahap implementasi, tahap verifikasi dan tahap pemeliharaan. Hasil teknik steganalysis dengan metode *chi square attack*, yaitu uji coba pertama menghasilkan nilai selisih pada ukuran kapasitas *file* sebesar 1 hingga 12 persen, yang membuktikan teknik dengan metode *chi square attack* dapat mengestimasi atau memperkirakan panjang pesan yang disisipi dalam suatu objek gambar. Pada tahap uji gambar acak diperoleh hasil *chi square* 0,60021 yang berarti 60 persen peluang adanya pesan yang disisipkan dalam gambar dan sebesar 0,09999 ataupun sebesar 0,10001 yang berarti 9 persen atau terdapat 10 persen peluang adanya pesan yang disisipkan atau dapat diartikan gambar tersebut tidak disisipi suatu pesan atau data.

Kata Kunci — Android Studio, *Chi-square attack*, Steganalysis, Steganografi, *Waterfall*

Abstract – *Steganography is the art of writing secret messages to hide information by inserting messages into an image, so it does not rule out steganography being used for criminal purposes or the insertion of pornographic files. While steganalysis is a technique to detect the presence of hidden messages on the cover image. In this study using steganalysis technique with the chi-square attack method to detect the presence of messages statistically. The purpose of this study is to develop a steganalysis technique using the chi-square attack method to detect JPEG stego-image format and find out the results of the analysis of the chi-square attack method applied to the Android Studio application. The method used in this study is the waterfall method, which consists of the needs analysis phase, the design phase, the implementation phase, the verification phase and the maintenance stage. The results of the steganalysis technique using the chi square attack method, which is the first trial, produce a difference in the size of the file capacity of 1 to 12 percent, which proves that the technique using the chi square attack method can estimate or estimate the length of messages inserted in an object image. In the random image test phase, the result of chi square is 0.60021 which means 60 percent chance of the message being inserted in the picture and 0.09999 or 0.10001 which means 9 percent or there is a 10 percent chance of the message being inserted or can be interpreted in the picture The message or data is not inserted*.

Keywords *— Android, Chi-square attack, Steganalysis, Steganography, Waterfall*

1. PENDAHULUAN

Seiring perkembangan Teknologi Informasi yang sangat pesat, ancaman terhadap keamanan data dan informasi pesan bersifat rahasia yang dibutuhkan sangat besar. Berbagai ancaman seperti tindakan penyadapan dari orang yang tidak bertanggung jawab. Dengan alasan itu, maka salah satu cara yang dapat digunakan untuk melindungi data atau sebuah pesan yaitu dengan terknik steganografi (Pamungkas, F.G, 2017). Teknik steganografi merupakan seni penyembunyian informasi dengan cara penyisipan pada suatu citra (Nugroho, D, 2016). Tidak menutup kemungkinan steganografi juga digunakan untuk penyisipan file berbau pornografi atau untuk tujuan kriminal. Banyak metode steganografi untuk melekatkan sejumlah besar pesan rahasia di dalam *pixel* pada *cover* atau penutup *image*, sehingga dengan adanya data tersembunyi tidak dapat didiagnosis. Maka sangat diperlukan deteksi untuk mengungkap keberadaan pesan dalam suatu citra yang dicurigai untuk tujuan kriminal atau yang berbau pornografi.

Steganalysis merupakan teknik untuk mendeteksi dan jika memungkinkan mengekstrak sekaligus menghancurkan pesan rahasia dari sebuah citra penampung atau *cover image*. Salah satu teknik steganografi yaitu LSB (*Least Significant Bit*) yang bekerja dengan mengganti bit-bit data yang tidak berguna atau nilai bit terendah dalam berkas citra penampung dan bit-bit informasi yang disembunyikan (*embedded message*).

Karenanya teknik LSB (*Least Significant Bit*) mengambil keuntungan dari keterbatasan indera manusia, maka akan dilakukan penyerangan pada *cover stego image* dengan menggunakan salah satu teknik steganalysis yaitu *chi-square attack* yang terdapat pada metode statistical *attack*.

Prinsip *chi-square attack* dalam mendeteksi keberadaan *embedded message* (Rokhman, N, 2011), peneliti memanfaatkan metode tersebut untuk mendeteksi pesan dalam media gambar berformat JPEG. Dimana selain mudah didapatkan di internet, gambar berformat JPEG merupakan gambar yang mengalami kompresi *lossy* atau menghilangkan beberapa data dari *cover image* dan menghasilkan gambar dengan ukuran kecil, sehingga dapat dengan mudah meningkatkan informasi yang tersimpan di data (Morkel, dkk, 2005).

Tujuan dari penelitian ini adalah menerapkan teknik steganalysis menggunakan metode *chi-square attack* untuk mendeteksi stego image berformat JPEG dan menganalisa keberadaan pesan yang tersembunyi dalam suatu citra dengan bantuan software Android Studio.

2. METODE PENELITIAN

Metode penelitian yang digunakan pada penelitian ini adalah metode air terjun atau yang sering disebut metode waterfall. Metode waterfall merupakan sebuah metode yang menggambarkan pendekatan secara sistematis dan juga berurutan untuk pengembangan sebuah perangkat lunak. Tahap dari metode waterfall adalah sebagai berikut:

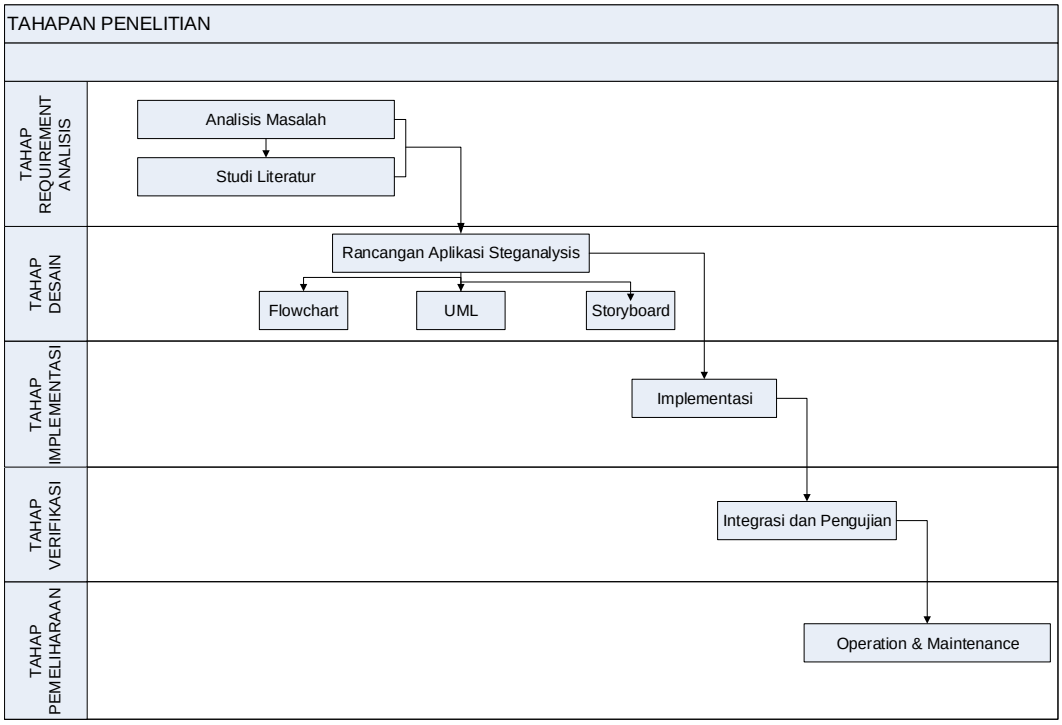
a. Tahap *Requirement Analysis*
   Penelitian ini dimulai dengan melakukan analisis kebutuhan terlebih dahulu, yang mencakup analisis terhadap masalah yang ada dan kemudian melakukan studi literatur, yang dimulai dengan pengumpulan data mengenai steganografi, steganalysis, metode *chi-square attack*, dan Android Studio.
b. Tahap Desain
   Setelah dilakukan tahap analisis kebutuhan, tahap selanjutnya yang akan dilakukan adalah tahap desain. Pada tahap desain akan menjelaskan gambaran skenario dari interaksi antara pengguna dan kegiatan yang dapat dilakukannya terhadap aplikasi.

3. Implementasi
   Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas
4. Verifikasi
   Pada tahap verifikasi, aplikasi akan diujicoba pada emulator dan perangkat *handphone*. Pada tahap uji pertama dilakukan peneliti tanpa ada peranserta pihak lain dengan masukan gambar yang sama yaitu, 3 gambar stego dan 3 gambar *non-stego* berformat JPEG. Setelah melakukan tahap uji pertama dilanjutkan dengan uji coba dengan beberapa masukan gambar acak berformat JPEG. Tahap ini menggunakan metode pengujian t-test yang ditujukan untuk membandingkan hasil beberapa sampel gambar yang telah di uji.

5. Maintenance

Tahap akhir dalam model waterfall. Perangkat lunak yang sudah jadi, dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

Langkah-langkah dalam penelitian *waterfall* ini, dapat dilihat pada gambar 1. berikut ini:

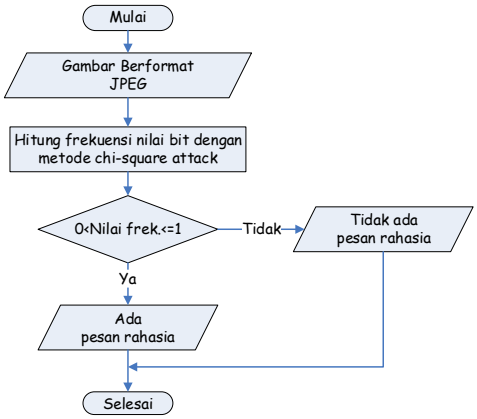


Gambar 1. Tahap Metode *Waterfall*

3. HASIL DAN PEMBAHASAN

3.1. Cara kerja Teknik stegalysis menggunakan metode chi square attack

Berikut adalah *flowchart* dan analisa pendeteksian *stego image* menggunakan metode *chi-square attack*:



Gambar 2. *Flowchart* pendeteksian gambar JPEG dengan *chi-square attack*

Analisa pendeteksian *stego image* berformat JPEG adalah menentukan nilai frekuensi pada penyisipan pesan acak, yaitu frekuensi aktual dan frekuensi yang diharapkan dengan menggunakan metode *chi-square attack*.

Jika hasil perbandingan nilai frekuensi aktual dan frekuensi yang diharapkan berbeda jauh dan mendekati 0 (nol) maka distribusi LSB tidak acak sehingga besar kemungkinan tidak ada pesan yang disisipkan dalam LSB. Sebaliknya jika sama sehingga mendekati nilai 1 (satu) maka distribusi LSB tersebut acak dan besar kemungkinan ada pesan yang disisipkan dalam LSB.

3.2. Implementasi perangkat lunak

Proses implementasi dimulai dengan membuat desain yang selanjutnya diubah menjadi interface aplikasi dengan menggunakan perangkat lunak Android Studio versi 3.3. Hasil *interface* aplikasi yang dibuat adalah sebagai berikut.

Bagian terpenting dari aplikasi *Chi Steganalysis Tool* yaitu pada menu Masukan, yaitu 1). Proses pengambilan gambar berformat JPEG, 2). Proses analisa *stego image*, sehingga dapat mendeteksi ada atau tidaknya pesan yang disisipkan dalam suatu objek gambar dengan hasil penghitungan *chi square attack*.

a. Berikut proses pengambilan gambar berformat JPEG :

```
Intent iGallery = new Intent(Intent.ACTION_PICK, MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
iGallery.setType("image/JPEG");
startActivityForResult(iGallery, REQUEST_IMAGE_GALLERY);
```

b. Berikut proses chi square attack dengan melakukan serangan dari titik atas ke bawah pada warna piksel (red):

```
for (j = 0; j < height; j++) {
    for (i = 0; i < width; i++) {
        if (block < chi.length) {
            red = (int) new Color().red();
            values[red]++;
            nbBytes++;
            if (nbBytes > size) {
                for (int k = 0; k < expectedValues.length; k++) {
                    expectedValues[k] = (values[2 * k] + values[2 * k + 1]) / 2;
                    pov[k] = values[2 * k];
                }
                chi[block] = new ChiSquareTest().chiSquareTest(expectedValues, pov);
                block++;
                nbBytes = 1;
                }
        }
```

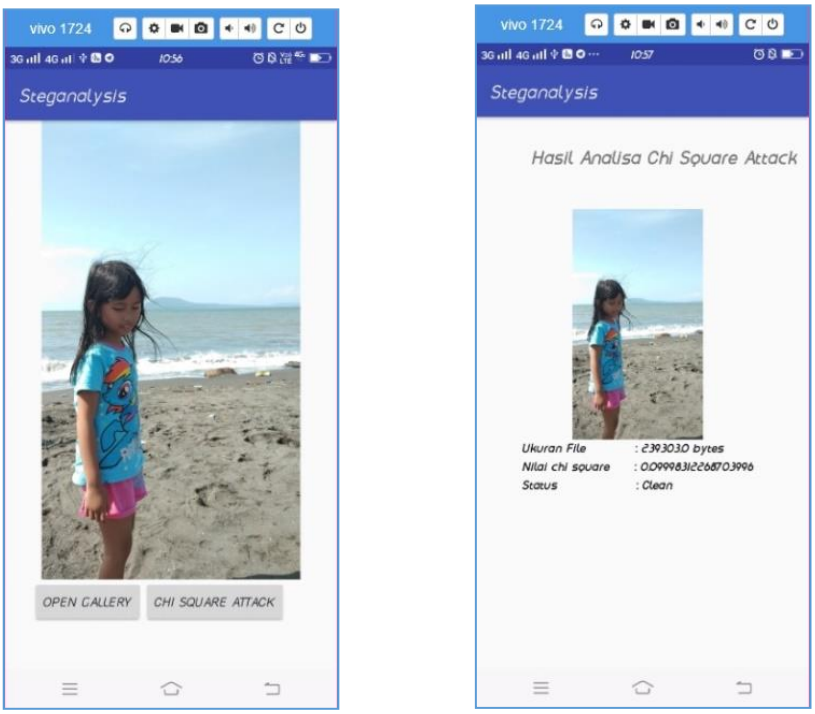
3.3. Uji coba pada perangkat Android

Pada pengujian pertama aplikasi Chi Steganalysis Tool pada perangkat Android dilakukan masukan 3 gambar tanpa proses steganografi dan 3 stego image (gambar dengan proses steganografi) berformat JPEG dengan gambar yang sama.

Salah satu pengujian aplikasi menggunakan masukan gambar tanpa proses steganografi dengan format JPEG, resolusi 585 x 1040 dan ukuran kapasitas gambar 233 KB yang ditunjukkan pada gambar 3.

Gambar 3. *Hidden Image*

Setelah memasukkan gambar *non-stego*, akan dilakukan proses steganalysis yaitu analisa stego image berformat JPEG dengan menggunakan metode *chi square attack*. Hasil dari analisa dengan menggunakan metode *chi square attack* ditampilkan pada gambar 4.



Gambar 4. Tampilan hasil *pick image* dan *analisa chi square attack*

Setelah mengetahui hasil analisa *chi square attack* pada gambar *non-stego*, selanjutnya akan dilakukan uji coba gambar dengan proses steganografi (*stego image*) pada gambar yang sama. Sedangkan *file* atau data yang disisipkan di dalam *cover image* sebesar 183 *bytes* ditunjukkan pada Gambar 5.

Gambar 5. *Pesan Teks*

Hasil pengujian pertama aplikasi *chi steganalysis tool* pada perangkat Android pada gambar tanpa proses steganografi dan gambar dengan proses steganografi (*stego image*) dapat ditunjukkan dalam tabel 1.

Tabel 1. Hasil pengujian *chi steganalysis tool* pada gambar *stego* dan gambar tanpa proses steganografi berformat JPEG dengan metode *chi square attack*

No.	Citra (.jpg)	Ukuran Kapasitas (*bytes*)	Nilai *chi square*	Status
1		239303	0,09998	*Clean*
2		239310	0,6	*Stego*
3		310299	0,1	*Clean*
4		310311	0,60015	*Stego*

Dari hasil pengujian tahap pertama, 2 gambar yang diambil dari *gallery device mobile* dapat dijelaskan secara berurutan dari gambar tanpa proses steganografi yang kemudian pengujian untuk gambar dengan proses steganografi. Pada Gambar 1 nilai ukuran kapasitas sebesar 239303 *bytes* dengan nilai *chi square* sebesar 0,09998. Sedangkan pada Gambar 2 dengan gambar yang sama, gambar dengan proses steganografi menghasilkan ukuran kapasitas sebesar 239309 bytes dengan nilai *chi square* sebesar 0,6. Pada pesan berukuran 138 *bytes* dapat dideteksi sebesar 7 *bytes* dan pengujian selanjutnya pada Gambar 3 yang dibandingkan dengan hasil pengujian Gambar 4 terdeteksi sebesar 12 *bytes* hingga sampel Gambar 5 dengan hasil pengujian Gambar 6 terdeteksi sebesar 1 *bytes*. Dari hasil 6 gambar tersebut terlihat selisih pada nilai kapasitas ukuran file sebesar 1 hingga 12 persen dari *cover image*.

Steganalysis dengan metode *chi square attack* dapat menunjukkan perkiraan panjang pesan dan angka yang akurat untuk penyisipan pesan secara berurutan atau sekuensial. Selisih yang terdeteksi pada ukuran kapasitas ini disebabkan oleh *bit-bit* pesan yang berfokus pada suatu bagian dan mengakibatkan distribusi *bit* pesan merata pada satu bagian yang membuat distribusi frekuensi PoV mendekati rata-rata. Apabila distribusi PoV mendekati rata-rata, maka akan menyerupai distribusi frekuensi yang diharapkan oleh sebuah citra yang telah disisipi pesan yaitu dengan nilai probabilitas diatas 0,5 atau nilainya mendekati angka 1. Hal ini terdapat pada uji coba Gambar 2 yang telah disisipi pesan dengan nilai *chi square* sebesar 0,6, Gambar 4 didapatkan nilai *chi square* sebesar 0,60015, dan Gambar 6 dengan nilai *chi square* sebesar 0,60059.

Setelah dilakukan uji coba tahap pertama pada perangkat Android, akan dilakukan uji coba pada 5 gambar acak berformat JPEG yang didapatkan dari internet. Berikut adalah hasil pengujian beberapa gambar acak berformat JPEG yang dianalisa dengan metode *chi square attack* yang ditunjukkan pada tabel 2.

Tabel 2. Hasil pengujian chi steganalysis tool pada gambar acak berformat JPEG dengan metode chi square attack

No.	Citra (.jpg)	Ukuran Kapasitas (bytes)	Nilai *chi square*	Status
1		47016	0,60021	*Stego*
2		104397	0,59999	*Stego*
3		143962	0,60039	*Stego*
4		80800	0,09999	*Clean*
5		82267	0,09991	*Clean*

4. KESIMPULAN

Kesimpulan yang didapat pada penelitian ini antara lain:

a. Teknik *steganalysis* dengan menggunakan metode *chi square attack* dapat diterapkan pada aplikasi berbasis Android dan menghasilkan aplikasi yang bernama *chi steganalysis tool* untuk mendeteksi *stego image* berformat JPEG. Dimana gambar dengan format JPEG selain mudah didapat pada internet juga mudah meningkatkan informasi yang tersimpan di data karena mengalami kompresi *lossy* atau menghilangkan beberapa data dari *cover image* dan menghasilkan gambar dengan ukuran kecil.
b. Hasil teknik steganalysis dengan metode *chi square attack* yaitu pada pengujian tahap pertama dengan sampel 2 gambar yang sama dengan proses yang berbeda, pertama gambar tanpa proses steganografi yang kedua gambar dengan proses steganografi terdapat selisih nilai ukuran kapasitas file sebesar 1 *bytes*, 7 *bytes* dan 12 *bytes*, yang membuktikan steganalysis dengan metode *chi square attack* dapat mengestimasi atau memperkirakan dengan akurat panjang pesan

yang disisipi secara sekuensial dalam suatu objek gambar. Gambar 1 dengan nilai *chi square* sebesar 0, 09987 yang berarti nilai probabilitas mendekati nilai 0 dan Gambar 2 dengan nilai *chi square* sebesar 0,60021 yang berarti nilai probabilitas lebih dari 0,5 atau mendekati 1, hal ini membuktikan distribusi bit pesan merata pada satu bagian yang membuat distribusi PoV mendekati rata-rata dan menyerupai distribusi frekuensi yang diharapkan oleh sebuah citra yang telah disisipi pesan.

c. Pada tahap pengujian 5 gambar acak yang diambil dari internet, metode *chi square attack* dapat melakukan deteksi gambar secara sekuensial atau berurutan yang dilihat dari hasil nilai *chi square* 0,60021 yang berarti 60 persen peluang adanya pesan yang disisipkan dalam objek gambar dan nilai *chi square* sebesar 0,09999 ataupun sebesar 0,10001 yang berarti 9 persen atau terdapat 10 persen peluang adanya pesan yang disisipkan dan dapat diartikan gambar tersebut tidak disisipi suatu pesan atau data.

DAFTAR PUSTAKA

[1] Christy, Atika Sari, dkk. 2016. Penyembunyian Data Untuk Seluruh Ekstensi File Menggunakan Kriptografi Vernam Chiper dan Bit Shiffting. *Jurnal of Applied Intelligent System*. Vol 1. No 3. Oktober 2016.

[2] Dedy, Abdullah, dkk. 2016. Implementasi Algoritma Blowfish dan Metode Least Significant Bit Insertion Pada Video MP4. *Jurnal Pseudocode*. Vol 3. No 2. September 2016.

[3] Friski, Gatra Pamungkas, dkk. 2017. Implementasi Teknik Steganalisis Menggunakan Metode Improvement Difference Image Histogram Pada Steganografi LSB. *Seminar Nasional Inovasi dan Aplikasi Teknologi Industri*. ITN Malang. 4 Pebruari 2017.

[4] Muhamad, Fitra Syawal, dkk. 2016. Implementasi Teknik Steganografi Menggunakan Vigenere Cipher dan Metode LSB. *Jurnal TICOM*. Vol 4. No 3. Mei 2016.

[5] Munir, Rinaldi. 2016. Eksperimen Steganalisis Dengan Metode Visual Attack Pada Citra Hasil Stego Berformat GIF. *Seminar Nasional Aplikasi Teknologi Informasi*. Yogyakarta. 6 Agustus 2016.