

# Aplikasi Steganografi dengan Menggunakan Metode F5

M. Barkah Akbar<sup>1</sup>, Edy Victor Haryanto<sup>2</sup>

<sup>1,2</sup>Universitas Potensi Utama

Jl. K.L. Yos Sudarso Km. 6,5 No. 3 A Tj. Mulia – Medan

## Abstrak

Steganografi merupakan seni menyembunyikan informasi pada suatu media sehingga keberadaannya tidak terdeteksi oleh pihak lain yang tidak berhak atas informasi tersebut. Citra digital adalah salah satu media yang dapat digunakan untuk menyisipkan pesan dalam teknik steganografi. Algoritma yang digunakan untuk penelitian ini adalah F5. Algoritma F5 menyisipkan bit pesan ke dalam bit koefisien DCT hasil kuantisasi yang telah dipermutasi. Penilaian sebuah algoritma steganografi yang baik salah satunya dapat dipandang dari banyaknya pesan yang dapat disisipkan dalam citra digital, serta waktu eksekusi yang diperlukan untuk penyisipan dan pengungkapan. Untuk mengetahui kelayakan F5 sebagai algoritma steganografi, maka akan dilakukan uji maksimum muatan dan uji waktu eksekusi. Dalam uji maksimum muatan ini, citra digital akan disisipi pesan hingga tidak bisa disisipi lagi, lalu didapatkanlah persentase muatan dari masing-masing jenis gambar sambil dihitung waktu eksekusi dari setiap penyisipan. Hasil dari uji muatan maksimum adalah jenis gambar grayscale dapat disisipi pesan paling banyak dibandingkan dengan jenis gambar lainnya. Dari hasil uji waktu eksekusi didapatkanlah ratio waktu eksekusi penyisipan dan pengungkapan pesan yaitu 1 : 6 dalam satuan detik.

**Kata kunci :** *Steganografi, DCT, Kuantisasi, Permutasi, F5*

## Abstract

Steganography is the art of hiding information on the media so that its presence is not detected by the other party is not entitled to that information. Digital images is one of the media that can be used to insert messages in steganographic techniques. The algorithm used for this study is F5. F5 bit algorithm insert messages into bit quantization DCT coefficients that have been permutation results. Steganography algorithms that assess both of them can be seen from the number of messages that can be incorporated in a digital image, as well as the execution time required for insertion and disclosure. To determine the feasibility of the F5 as steganography algorithms, it will be a test load and test execution maximum time. In this maximum test load, a digital image will be inserted until the message cannot be inserted again, and then earn a percentage cost of each type of temporary image calculated execution time of each insertion. The maximum load test result is the kind of Grayscale images can be inserted messages compared to most other image types. From the test results obtained ratio of execution time execution time insertion and disclosure of message 1: 6 in seconds.

**Keywords :** *steganography , DCT , quantization , permutations , F5*

## 1. PENDAHULUAN

Steganografi adalah seni dan ilmu komunikasi tersembunyi mirip dengan kriptografi. Steganografi memungkinkan dua pihak yang saling percaya untuk bertukar pesan secara rahasia. Perbedaan steganografi dan kriptografi adalah steganografi menambahkan lapisan perlindungan dengan menyembunyikan pesan tersebut kedalam penampung yang terlihat [2]. Sejalan dengan perkembangan teknologi informasi saat ini, penyembunyian pesan menggunakan *invisible ink* dan kertas sebagai penampung telah tergantikan media digital seperti *image*, audio ataupun video [3]. Terdapat banyak metode steganografi yang lemah terhadap serangan visual dan statistik. Tetapi ada juga metode yang tahan terhadap serangan visual dan statistik dengan kapasitas yang kecil [11].

Adapun tujuan dari penelitian ini adalah merancang sebuah aplikasi yang dapat menyisipkan pesan ke dalam sebuah citra JPEG dengan menggunakan metode F5, serta mengungkapkan kembali pesan tersebut dari hasil penyisipan sebelumnya.

## 2. METODE PENELITIAN

Adapun yang menjadi langkah-langkah metodologi yang akan digunakan dalam proses penyelesaian tugas akhir ini adalah sebagai berikut :

1. Mempelajari masalah-masalah yang berhubungan dengan algoritma F5 seperti kompresi citra, transformasi citra, kuantisasi citra, permutasi citra, steganografi, generator bilangan acak semu, dan steganografi.
2. Mengumpulkan data untuk mendapatkan tambahan materi dan mempelajari informasi dari buku, jurnal-jurnal, artikel, literatur perpustakaan dan sumber lainnya yang membahas tentang algoritma F5 seperti dijelaskan sebelumnya, yang kemudian digunakan sebagai dasar acuan dalam menyelesaikan masalah pada tugas akhir yang dikerjakan.
3. Melakukan analisis untuk menentukan kebutuhan fungsional dan non fungsional perangkat lunak steganografi, seperti fitur embed (penyisipan) pesan dan ekstrak (pengungkapan) kembali pesan tersebut menggunakan metode F5.
4. Merancang perangkat steganografi dengan fitur berdasarkan analisis yang dilakukan sebelumnya menggunakan visual basic 2008.

Melakukan pengujian untuk menentukan banyaknya pesan yang dapat disisipkan kedalam suatu citra, jenis citra yang baik untuk steganografi serta rasio waktu eksekusi antar embed dan ekstrak

## 3. HASIL DAN PEMBAHASAN

### 3.1. Proses Discrete Cosine Transform (DCT)

Proses transformasi DCT dimulai dengan membagi citra menjadi beberapa blok dengan ukuran 8 x 8 *pixel*/ blok, blok-blok *pixel* tersebut masing-masing diproses menjadi 64 koefisien DCT. Berikut adalah rumus yang digunakan untuk memproses 1 blok 8 x 8 *pixel* dengan DCT :

$$F(u,v) = \frac{C(u)C(v)}{4} \sum_{i=0}^7 \sum_{j=0}^7 \cos \frac{(2i+1)u\pi}{16} \cos \frac{(2j+1)v\pi}{16} f(i,j) \quad \dots\dots\dots (1)$$

Keterangan :

$f(i, j)$  = matrix input dengan panjang  $i$ , dengan lebar  $j$

$F(u, v, )$  = hasil proses dct dengan panjang  $u$ , dan lebar  $v$

DCT hanya dapat memproses koefisien yang nilainya diantara -127 dan 127. Oleh karenanya, masing-masing koefisien dari matrix sebelumnya dikurangkan 128 sehingga mendapatkan hasil sebagai berikut :

62	26	31	26	-33	25	68	2
4	27	-34	35	30	-100	77	21
127	60	-96	-127	-63	-49	127	-86
109	-92	16	-81	-21	-55	19	-52
21	-101	16	-63	-80	-116	-48	-44
58	67	127	-18	-37	-62	-51	-92
127	-5	-68	-77	27	65	105	62
82	-61	-65	0	25	-36	-85	-45

Berikut Perhitungannya :

tmpdct(0,0) = 0 + (cos((2\*0+1)\*0\*Phi)/16) \* (cos((2\*0+1)\*0\*Phi)/16) \* 62 = 62  
 tmpdct(0,1) = 62 + (cos((2\*0+1)\*0\*Phi)/16) \* (cos((2\*1+1)\*0\*Phi)/16) \* 26 = 88  
 tmpdct(0,2) = 88 + (cos((2\*0+1)\*0\*Phi)/16) \* (cos((2\*2+1)\*0\*Phi)/16) \* 31 = 119

tmpdct(7,5) = -197 + (cos((2\*7+1)\*0\*Phi)/16) \* (cos((2\*5+1)\*0\*Phi)/16) \* -36 = -233  
 tmpdct(7,6) = -233 + (cos((2\*7+1)\*0\*Phi)/16) \* (cos((2\*6+1)\*0\*Phi)/16) \* -85 = -318  
 tmpdct(7,7) = -318 + (cos((2\*7+1)\*0\*Phi)/16) \* (cos((2\*7+1)\*0\*Phi)/16) \* -45 = -363

hasildct(0,0) = 0.707106781186547 \* 0.707106781186547 / 4 \* -363.00 = -45.38  
 hasildct(0,1) = 0.707106781186547 \* 1 / 4 \* 676.86 = 119.65  
 hasildct(0,2) = 0.707106781186547 \* 1 / 4 \* 949.92 = 167.92

$$\begin{aligned} \text{hasildct}(7,5) &= 1 * 1/4 * 59.85 = 14.96 \\ \text{hasildct}(7,6) &= 1 * 1/4 * -107.57 = -26.89 \\ \text{hasildct}(7,7) &= 1 * 1/4 * -88.28 = -22.07 \end{aligned}$$

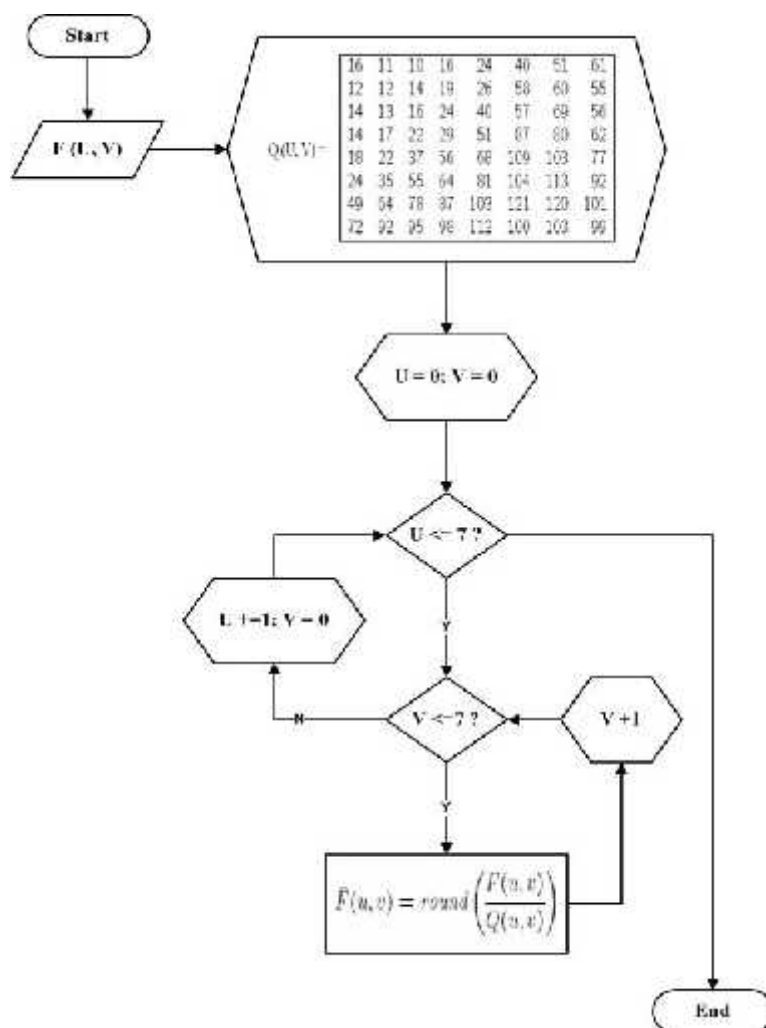
Hasil dari proses DCT adalah :

-45	120	168	102	22	117	-33	122
42	-33	26	-39	-60	46	-99	-1
125	-65	-38	29	46	-39	-41	-87
55	37	-60	-126	-57	-9	67	-18
-92	5	-61	-29	70	37	111	-2
103	-49	89	123	-57	22	37	8
-28	139	-7	7	-82	0	4	-20
-29	-67	94	68	-23	15	-27	-22

Di dalam masing-masing blok koefisien tersebut, koefisien DCT pertama dari setiap blok disebut koefisien DC, sedangkan sisanya adalah koefisien AC. Koefisien DC pada 1 blok menyimpan informasi dari seluruh koefisien AC dalam satu blok, oleh karenanya umumnya koefisien DC biasanya memiliki nilai yang paling besar dibanding koefisien AC yang lainnya, walaupun dalam beberapa kasus nilai Koefisien DC dapat lebih kecil dibandingkan Koefisien AC. Koefisien DC cenderung memiliki nilai yang paling besar di banding koefisien AC bila matrix input memiliki nilai yang tidak beragam.

### 3.2. Proses Kuantisasi

Proses kuantisasi dimulai dengan membagi setiap nilai *pixel* hasil DCT dengan setiap nilai konstanta elemen matriks kuantisasi, dimana nilai-nilai setiap elemen matriks tersebut menentukan kualitas hasil kompresi. Adapun tahapan proses kuantisasi dapat dilihat pada Gambar 3 dibawah ini.



Gambar 3. Flowchart Kuantisasi

Berikut adalah rumus yang digunakan untuk memproses 1 blok 8 x 8 *pixel* dengan Kuantisasi :

$$\hat{F}(u, v) = \text{round} \left( \frac{F(u, v)}{Q(u, v)} \right) \dots\dots\dots (2)$$

Keterangan :

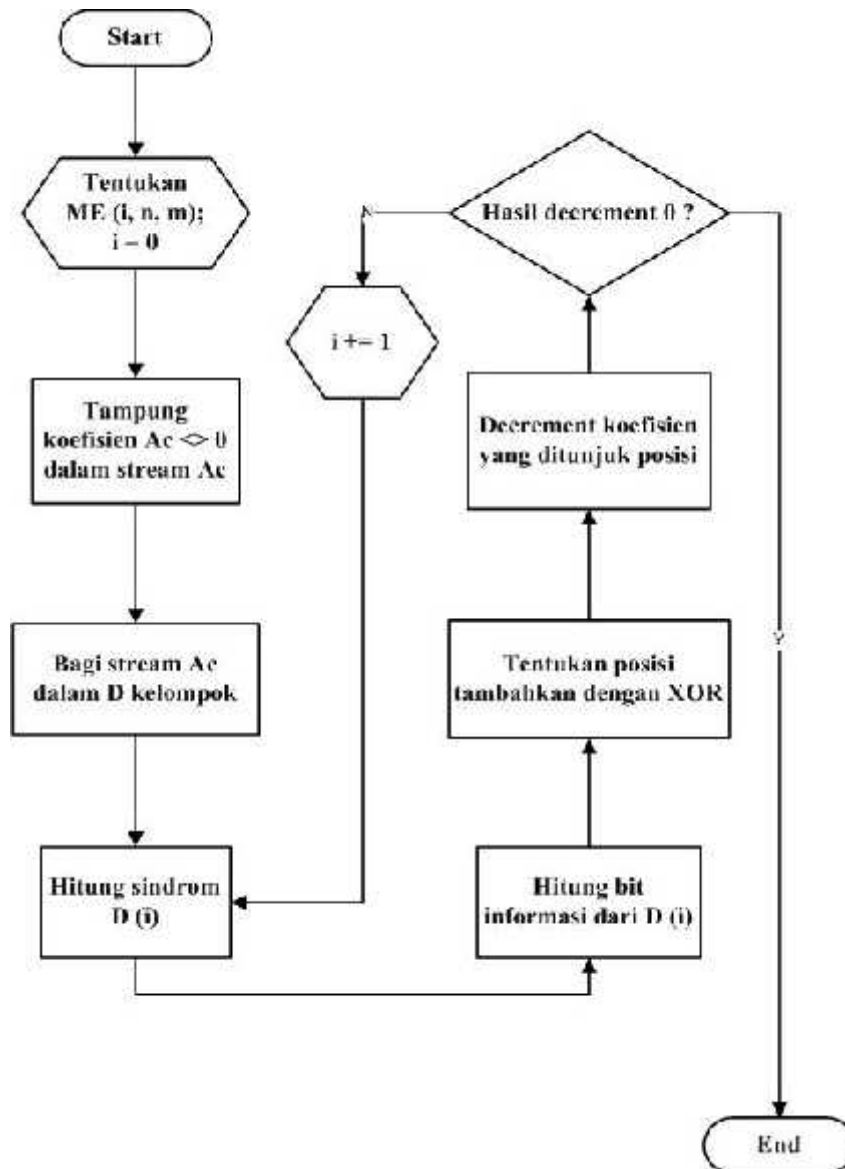
$F(u, v)$  = matrix input dengan panjang  $i$ , dengan lebar  $j$

$Q(u, v)$  = matrix kuantisasi dengan panjang  $i$ , dengan lebar  $j$

$F'(u, v)$  = hasil proses kuantisasi dengan panjang  $u$ , dan lebar  $v$

### 3.3. Penyisipan dan Pengungkapan dengan Metode F5

Adapun cara kerja dari metode F5 dapat dilihat dari gambar 4 dibawah ini :



Gambar 4. Flowchart F5

Untuk menyisipkan pesan menggunakan F5 :

1. Tentukan Skema Matrix Embedding (n,m).
  1. Dimana i adalah banyak perubahan dalam 1 kelompok.
  2. n adalah banyak stream dalam 1 kelompok ;  $n = 2^m - 1$ .
  3. m adalah banyak bit pesan yang akan disisipkan dalam 1 kelompok.
2. Tampung stream koefisien AC yang tidak bernilai 0(h1).
3. Hitung banyak karakter maksimal yang dapat disisipkan dengan cara : h1 bagikan dengan n kalikan dengan m bagikan lagi dengan 8(banyak bit dalam 1 karakter)
4. Bila banyak karakter yang disisipkan melebihi banyak karakter maksimal yang bisa disisipkan, keluarkan pesan error, keluar dari penyisipan.
5. Isi 1 kelompok berdasarkan besar n kedalam D.
6. Tentukan bit informasi B dari D dengan ketentuan.

$$b_{(i,j)} = \begin{cases} c_{(i,j)} \bmod 2 & \text{if } c_{(i,j)} > 0 \\ |(c_{(i,j)} - 1) \bmod 2| & \text{if } c_{(i,j)} < 0 \end{cases} \dots\dots\dots (3)$$

7. Hitung Syndrome dengan fungsi Xor.

8. Tentukan posisi bit yang akan di ganti.
9. Decrement koefisien yang ditunjuk posisi.
10. Bila hasil bit yang telah terganti menjadi 0 maka ulangi langkah 3 dengan kelompok selanjutnya.

Asumsi pesan yang akan disisipkan (M) adalah (1 0 1), maka Skema Matrix Embedding adalah (7,3), Stream koefisien DCT (D) dan bit informasinya (S) adalah (3, -1, 2, 1, 1, -1, -3) dan (1, 0, 0, 1, 1, 0, 0).

$$P = \{p_1, p_2, p_3\}$$

$$S = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$$

1. Hitung syndrome  $O = \{o_1, o_2, o_3\}$ 

$$o_1 = \text{xor}(a_4, a_5, a_6, a_7)$$

$$o_2 = \text{xor}(a_2, a_3, a_6, a_7)$$

$$o_3 = \text{xor}(a_1, a_3, a_5, a_7)$$
2. Posisi (P) bit yang akan diubah dapat ditentukan dengan
 
$$P = \text{xor}(O, M)$$

Untuk lebih jelasnya tentang metode F5, dapat dilihat dari contoh penyisipan dan pengungkapan pesan dibawah ini :

Berikut adalah Stream Koefisien DCT :

-3,(1, -2, 0, 0, 2, -2, -3, 6, -2),(0, 0, 0, 17, -1, -1, -1, 11, -3, 1),(1, 0, -2, -1, -1, 1, 4, 1),(2, 0, -1, -2, 1, -1, 0, 2, 0, -5), (-4, -1, 0, 9, 0, 0, 0, 0, -5, -1, -1, 1),(4, 2, 1, 4, 2, 0, 0, 0, 1, 3), 0, 2, 0, -2.

D1 = (1,-2,2,-2,-3,6,-2) ;(1, -2, 0, 0, 2, -2, -3, 6, -2)  
 D2 = (17,-1,-1,-1,11,-3,1) ;(0, 0, 0, 17, -1, -1, -1, 11, -3, 1)  
 D3 = (1,-2,-1,-1,1,4,1) ;(1, 0, -2, -1, -1, 1, 4, 1)  
 D4 = (2,-1,-2,1,-1,2,-5) ;(2, 0, -1, -2, 1, -1, 0, 2, 0, -5)  
 D5 = (-4,-1,9,-5,-1,-1,1) ;(-4, -1, 0, 9, 0, 0, 0, 0, -5, -1, -1, 1)  
 D6 = (4,2,1,4,2,1,3) ;(4, 2, 1, 4, 2, 0, 0, 0, 1, 3)

Pesan yang akan diinput : M(001)

Matrix Encoding Skema :  $(M, 2^m - 1) : (3, 8 - 1) : (3, 7)$

#### 11. Penyisipan pada D1

D1(1,-2,2,-2,-3,6,-2) ; S1(1,1,0,1,0,0,1)

#### 12. Dapatkan Syndrome

$O_1 = \text{Xor}(a_4, a_5, a_6, a_7) = \text{Xor}(1, 0, 0, 1) = 0$   
 $O_2 = \text{Xor}(a_2, a_3, a_6, a_7) = \text{Xor}(1, 0, 0, 1) = 0$   
 $O_3 = \text{Xor}(a_1, a_3, a_5, a_7) = \text{Xor}(1, 0, 0, 1) = 0$

#### 13. Dapatkan Posisi

$P \text{ xor}(M, O) = \{\text{Xor}(0, 0), \text{Xor}(0, 0), \text{Xor}(1, 0)\} = \{0, 0, 1\}$   
 $P = 001 = 1$

#### 14. Proses Index berdasarkan posisi dengan decrement nilai absolutnya

D2(0,-2,2,-2,-3,6,-2) ; S2(0,1,0,1,0,0,1)

Karena index menjadi 0 maka peristiwa ini dinamakan *Shrinkage*, maka pesan disisipkan pada kelompok D selanjutnya.

#### 15. Penyisipan pada D2

D2 (17,-1,-1,-1,11,-3,1) ; S2(1,0,0,0,1,0,1)

#### 1. Dapatkan Syndrome

$O_1 = \text{Xor}(a_4, a_5, a_6, a_7) = \text{Xor}(0, 1, 0, 1) = 0$   
 $O_2 = \text{Xor}(a_2, a_3, a_6, a_7) = \text{Xor}(0, 0, 0, 1) = 1$   
 $O_3 = \text{Xor}(a_1, a_3, a_5, a_7) = \text{Xor}(1, 0, 1, 1) = 1$

#### 2. Dapatkan Posisi

$P \text{ xor}(M, O) = \{\text{Xor}(0, 0), \text{Xor}(0, 1), \text{Xor}(1, 1)\} = \{0, 1, 0\}$

$$P = 010 = 2$$

3. Proses Index berdasarkan posisi dengan decrement nilai absolutnya.

D2(17,1,-1,-1,11,-3,1) ; S2(1,1,0,0,1,0,1)

Output :

-3,(1, -2, 0, 0, 2, -2, -3, 6, -2),(0, 0, 0, 17, 1, -1, -1, 11, -3, 1),(1, 0, -2, -1, -1, 1, 4, 1),( 2, 0, -1, -2, 1, -1, 0, 2, 0, -5), (-4, -1, 0, 9, 0, 0, 0, 0, -5, -1, -1, 1),(4, 2, 1, 4, 2, 0, 0, 0, 1, 3), 0, 2, 0, -2.

Karena Index Tidak Menjadi 0 maka Embed Sukses.

### 3.4. Pengungkapan Pesan dengan Metode F5

Stream Hasil Penyisipan

-3,(1, -2, 0, 0, 2, -2, -3, 6, -2),(0, 0, 0, 17, 1, -1, -1, 11, -3, 1),(1, 0, -2, -1, -1, 1, 4, 1),( 2, 0, -1, -2, 1, -1, 0, 2, 0, -5), (-4, -1, 0, 9, 0, 0, 0, 0, -5, -1, -1, 1),(4, 2, 1, 4, 2, 0, 0, 0, 1, 3), 0, 2, 0, -2.

D1 = (1,-2,2,-2,-3,6,-2)	;(1, -2, 0, 0, 2, -2, -3, 6, -2)
D2 = (17,1,-1,-1,11,-3,1)	;(0, 0, 0, 17, 1, -1, -1, 11, -3, 1)
D3 = (1,-2,-1,-1,1,4,1)	;(1, 0, -2, -1, -1, 1, 4, 1)
D4 = (2,-1,-2,1,-1,2,-5)	;( 2, 0, -1, -2, 1, -1, 0, 2, 0, -5)
D5 = (-4,-1,9,-5,-1,-1,1)	;( -4, -1, 0, 9, 0, 0, 0, 0, -5, -1, -1, 1)
D6 = (4,2,1,4,2,1,3)	;(4, 2, 1, 4, 2, 0, 0, 0, 1, 3)
D2 (17,1,-1,-1,11,-3,1)	; S2(1,0,0,0,1,0,1)

$$O_1 = \text{Xor}(a_4, a_5, a_6, a_7) = \text{Xor}(0, 1, 0, 1) = 0$$

$$O_2 = \text{Xor}(a_2, a_3, a_6, a_7) = \text{Xor}(1, 0, 0, 1) = 0$$

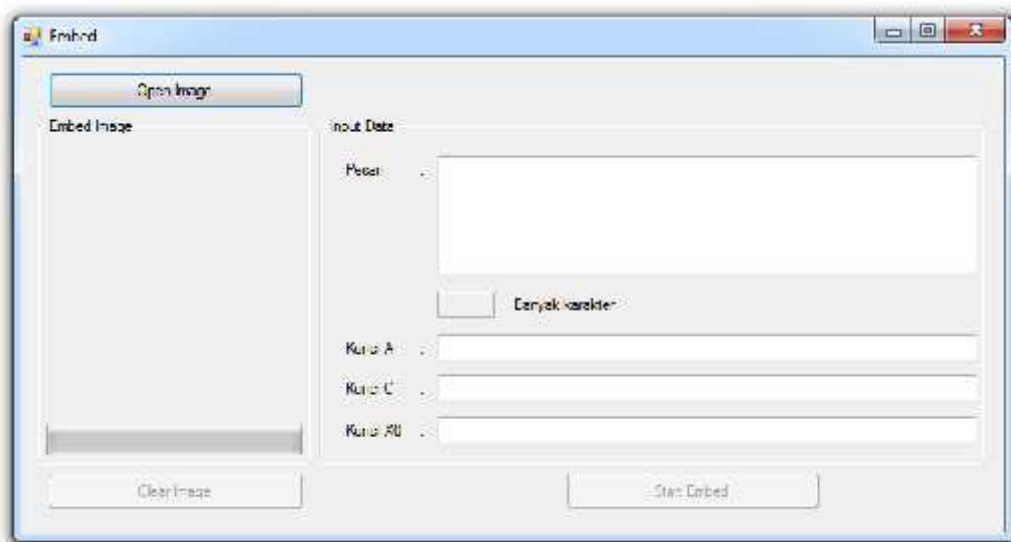
$$O_3 = \text{Xor}(a_1, a_3, a_5, a_7) = \text{Xor}(1, 0, 1, 1) = 1$$

$$M = O$$

$$M = 001.$$

### 3.5. Tampilan Program

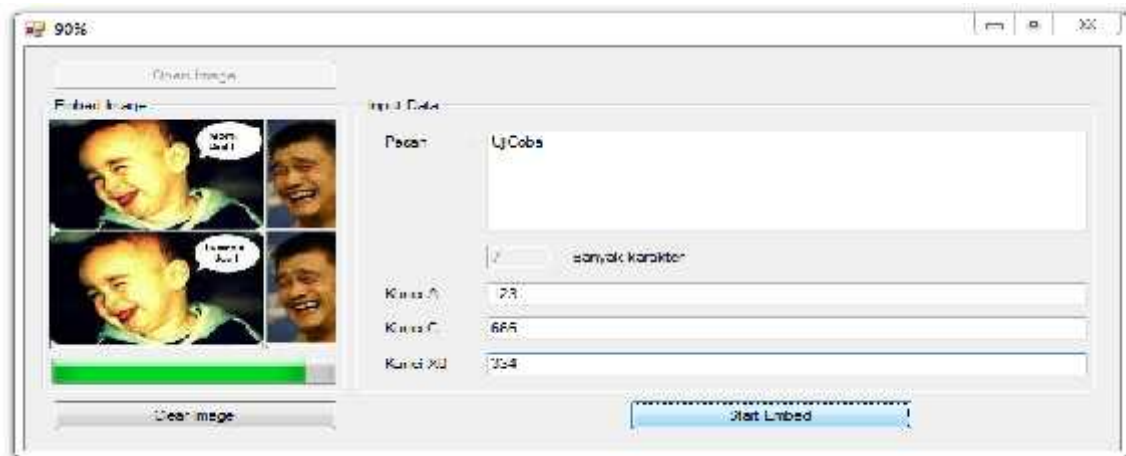
Menu *Embed* berfungsi untuk menyisipkan pesan ke dalam gambar Gambar 5 adalah tampilan menu *Embed* yang harus diisi oleh user.



Gambar 5. Tampilan Form *Embed*

Untuk dapat memulai *embed*, user diharuskan menginput kunci, pesan, serta *image*. Terdapat beberapa ketentuan dalam menginput, seperti :

1. Kunci harus lebih besar dari 0, kunci harus angka, khusus untuk kunci X0, besarnya kunci dibatasi berdasarkan *image* yang diinput.
2. Pesan dapat diinput dengan karakter huruf dan angka, namun banyaknya pesan dibatasi berdasarkan *image* yang diinput.
3. Ekstensi *image* yang dapat diinput adalah \*.jpg, \*.jpeg, dan \*.bmp.
4. Untuk ukuran *image*, ketentuannya adalah panjang *image* harus sama dengan lebar *image*, panjang atau lebar *image* harus habis dibagi dengan 8, luas *image* harus habis dibagi dengan 64.

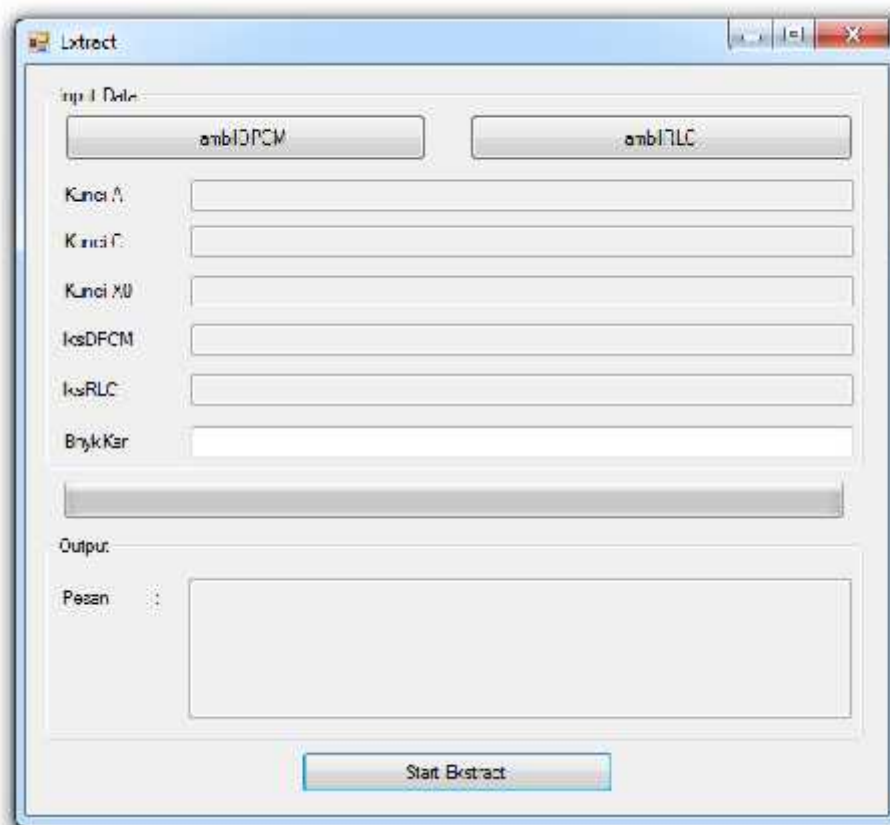


Gambar 6. Tampilan Proses Penyisipan Pesan

Setelah kunci, pesan, serta *image* selesai di-input, user dapat menekan tombol *start embed* untuk menyisipkan pesan. Proses penyisipan terlihat seperti pada gambar 6. Setelah proses penyisipan selesai, maka akan muncul message box yang menampilkan lamanya waktu yang dibutuhkan untuk proses penyisipan tersebut. Hasil dari proses embed adalah 2 file berekstensi \*.bin yang masing-masing memiliki nama dpcm.bin dan rlc.bin. kedua file tersebut disimpan kedalam folder hsl.

Setelah proses penyisipan (*embed*) selesai, user dapat mengungkap (*extract*) kembali pesan yang telah disisipkan pada menu extract seperti Gambar 7.





Gambar 7. Tampilan Menu Extract

Apabila *user* ingin mengungkapkan kembali pesan, langkah pertama yang harus dilakukan adalah *user* harus menginput file dpcm.bin dan rlc.bin dengan cara klik tombol ambilDPCM dan tombol ambilRLC, lalu cari posisi disimpan file dpcm.bin dan rlc.bin.

Setelah hasil *embed* diinput, untuk mendapatkan pesan sesuai dengan yang disisipkan, *user* harus memasukkan kunci serta banyak karakter sama seperti yang diinput pada saat proses *embed*. Tekan tombol “Start Embed” untuk memulai proses ekstraksi.

Pada pengujian yang pertama, gambar yang digunakan adalah gambar yang berukuran 400x400 *pixel*. Kunci A = 1, kunci C = 1, dan kunci X0 = 1. Pesan yang disisipkan kedalam gambar adalah karakter “a” berulang sebanyak panjang pesan. Dengan panjang pesan terdiri dari 1%, 5%, 10%, 15%, 20%, 25% dan 30% dari banyak pesan maksimal yang dapat disisipkan. Banyak pesan maksimal yang dapat disisipkan pada gambar 400x400px adalah 7500 karakter. Hasil dari pengujian terlihat pada tabel 1 berikut :

Tabel 1. Hasil Embed dan Extract pada Citra 400x400

Nomor Pengujian	Nama Gambar Uji	Tipe Gambar	Format Gambar	Persen Dari Maksimal Karakter Input	Banyak Karakter	Waktu Embed	Hasil Uji	Waktu Ekstrak
1	400putih	Monotone	jpg	1 %	75	-	OverLoad	-
2	400putih	Monotone	jpg	5 %	375	-	OverLoad	-
3	400putih	Monotone	jpg	10 %	750	-	OverLoad	-
4	400putih	Monotone	jpg	15 %	1125	-	OverLoad	-
5	400putih	Monotone	jpg	20 %	1500	-	OverLoad	-
6	400putih	Monotone	jpg	25 %	1875	-	OverLoad	-
7	400putih	Monotone	jpg	30 %	2250	-	OverLoad	-
8	400grayscale	Grayscale	bmp	1 %	75	5.7	Berhasil	19.7
9	400grayscale	Grayscale	bmp	5 %	375	5.179	Berhasil	20.841
10	400grayscale	Grayscale	bmp	10 %	750	5.38	Berhasil	35.146
11	400grayscale	Grayscale	bmp	15 %	1125	5.194	Berhasil	58.734

12	400grayscale	Grayscale	bmp	20 %	1500	5.538	Berhasil	92.227
13	400grayscale	Grayscale	bmp	25 %	1875	-	OverLoad	-
14	400grayscale	Grayscale	bmp	30 %	2250	-	OverLoad	-
15	400warna	Warna	jpg	1 %	75	3.73	Berhasil	5.772
16	400warna	Warna	jpg	5 %	375	3.151	Berhasil	9.75
17	400warna	Warna	jpg	10 %	750	3.182	Berhasil	23.181
18	400warna	Warna	jpg	15 %	1125	-	OverLoad	-
19	400warna	Warna	jpg	20 %	1500	-	OverLoad	-
20	400warna	Warna	jpg	25 %	1875	-	OverLoad	-
21	400warna	Warna	jpg	30 %	2250	-	OverLoad	-
22	400ksg	Monotone OTB	bmp	1 %	75	2.9	Berhasil	0.889
23	400ksg	Monotone OTB	bmp	5 %	375	-	OverLoad	-
24	400ksg	Monotone OTB	bmp	10 %	750	-	OverLoad	-
25	400ksg	Monotone OTB	bmp	15 %	1125	-	OverLoad	-
26	400ksg	Monotone OTB	bmp	20 %	1500	-	OverLoad	-
27	400ksg	Monotone OTB	bmp	25 %	1875	-	OverLoad	-
28	400ksg	Monotone OTB	bmp	30 %	2250	-	OverLoad	-
29	400brisi	Monotone OB	bmp	1 %	75	1.903	Berhasil	0.764
30	400brisi	Monotone OB	bmp	5 %	375	-	OverLoad	-
31	400brisi	Monotone OB	bmp	10 %	750	-	OverLoad	-
32	400brisi	Monotone OB	bmp	15 %	1125	-	OverLoad	-
33	400brisi	Monotone OB	bmp	20 %	1500	-	OverLoad	-
34	400brisi	Monotone OB	bmp	25 %	1875	-	OverLoad	-
35	400brisi	Monotone OB	bmp	30 %	2250	-	OverLoad	-

Dari pengujian pertama, rangkuman yang didapat adalah rata-rata waktu embed untuk citra berukuran 400x400 px adalah 4.18 detik sedangkan untuk waktu extract untuk citra berukuran 400x400 px adalah 26.7 detik, maka didapatlah ratio rata-rata waktu embed dan extract untuk citra 400x400 px adalah 1: 6.

Pada pengujian yang kedua, gambar yang digunakan adalah gambar yang berukuran 320x320 *pixel*. Kunci A = 1, kunci C = 1, dan kunci X0 = 1. Pesan yang disisipkan kedalam gambar adalah karakter “a” berulang sebanyak panjang pesan. Dengan panjang pesan terdiri dari 1%, 5%, 10%, 15%, 20%, 25% dan 30% dari banyak karakter maksimal yang dapat disisipkan. Banyak pesan maksimal yang dapat disisipkan pada gambar 320x320px adalah 4800 karakter. Hasil dari pengujian terlihat pada tabel 2.berikut :

Tabel 2. Hasil Embed dan Extract pada Citra 320x320

Nomor Pengujian	Nama Gambar Uji	Tipe Gambar	format	Persen Dari Maksimal Karakter Input	Banyak Karakter	Waktu Embed	Hasil Uji	waktu extract
1	320putih	Monotone	jpg	1 %	48	-	OverLoad	-
2	320putih	Monotone	jpg	5 %	240	-	OverLoad	-
3	320putih	Monotone	jpg	10 %	480	-	OverLoad	-
4	320putih	Monotone	jpg	15 %	720	-	OverLoad	-
5	320putih	Monotone	jpg	20 %	960	-	OverLoad	-
6	320putih	Monotone	jpg	25 %	1200	-	OverLoad	-
7	320putih	Monotone	jpg	30 %	1440	-	OverLoad	-
8	320grayscale	Grayscale	bmp	1 %	48	2.324	Berhasil	6.177
9	320grayscale	Grayscale	bmp	5 %	240	2.731	Berhasil	7.768
10	320grayscale	Grayscale	bmp	10 %	480	2.698	Berhasil	12.854
11	320grayscale	Grayscale	bmp	15 %	720	2.823	Berhasil	21.543
12	320grayscale	Grayscale	bmp	20 %	960	2.964	Berhasil	35.302
13	320grayscale	Grayscale	bmp	25 %	1200	-	OverLoad	-
14	320grayscale	Grayscale	bmp	30 %	1440	-	OverLoad	-
15	320warna	Warna	jpg	1 %	48	1.778	Berhasil	3.132
16	320warna	Warna	jpg	5 %	240	1.951	Berhasil	4.602
17	320warna	Warna	jpg	10 %	480	2.43	Berhasil	9.921
18	320warna	Warna	jpg	15 %	720	2.246	Berhasil	18.954
19	320warna	Warna	jpg	20 %	960	-	OverLoad	-

20	320warna	Warna	jpg	25 %	1200	-	OverLoad	-
21	320warna	Warna	jpg	30 %	1440	-	OverLoad	-
22	320ksg	Monotone OTB	bmp	1 %	48	1.294	Berhasil	0.452
23	320ksg	Monotone OTB	bmp	5 %	240	-	OverLoad	-
24	320ksg	Monotone OTB	bmp	10 %	480	-	OverLoad	-
25	320ksg	Monotone OTB	bmp	15 %	720	-	OverLoad	-
26	320ksg	Monotone OTB	bmp	20 %	960	-	OverLoad	-
27	320ksg	Monotone OTB	bmp	25 %	1200	-	OverLoad	-
28	320ksg	Monotone OTB	bmp	30 %	1440	-	OverLoad	-
29	320brisi	Monotone OB	bmp	1 %	48	1.263	Berhasil	0.405
30	320brisi	Monotone OB	bmp	5 %	240	-	OverLoad	-
31	320brisi	Monotone OB	bmp	10 %	480	-	OverLoad	-
32	320brisi	Monotone OB	bmp	15 %	720	-	OverLoad	-
33	320brisi	Monotone OB	bmp	20 %	960	-	OverLoad	-
34	320brisi	Monotone OB	bmp	25 %	1200	-	OverLoad	-
35	320brisi	Monotone OB	bmp	30 %	1440	-	OverLoad	-

Dari pengujian kedua, didapatkan rangkuman rata-rata waktu *embed* untuk citra ukuran 320x320 px adalah 2.22 detik, rata-rata waktu *extract* untuk citra ukuran 320x320 px adalah 11.01 detik, sehingga ratio rata-rata waktu *embed* dan *extract* untuk citra 320x320 px adalah 1 :5. Berdasarkan pengujian yang dilakukan sebelumnya, didapatlah hasil pengujian sebagai berikut :

1. Rata-rata rasio kompresi antara waktu embed dan waktu ekstrak dari semua ukuran citra adalah 5 : 32 atau 1 : 6.
2. Urutan citra dengan rata-rata muatannya paling besar adalah Citra Grayscale(mencapai 30%) > Citra Berwarna(mencapai 30%)> Citra monoton dengan objek tak berbidang(mencapai 10%) > Citra monoton dengan objek berbidang(mencapai 5%) > Citra monoton(tidak mencapai 1%).
3. Citra dengan objek tidak berbidang dan citra dengan objek berbidang memiliki rata-rata muatan yang sama untuk setiap ukurannya.
4. Ukuran citra berbanding lurus dengan banyaknya pesan yang dapat disisipkan, semakin besar ukuran citra maka semakin banyak pesan yang dapat disisipkan.

#### 4. KESIMPULAN

Rasio waktu eksekusi rata – rata antara proses *embed* dan proses *extract* adalah 1 : 6 dalam satuan detik. *Stego image* dengan kapasitas penyisipan paling banyak adalah *stego image* dengan jenis *Grayscale* dimana mencapai 30% dari kapasitas penyisipan maksimum, sementara *stego image* dengan kapasitas penyisipan paling sedikit adalah *stego image* dengan jenis *Monotone* yang bahkan tidak mencapai 1% dari kapasitas penyisipan maksimum. *Stego image* monoton dengan object tidak berbidang memiliki kapasitas penyisipan yang relative sama dengan *stego image* monoton dengan objek berbidang. Ukuran *Stego image* berbanding lurus dengan banyaknya pesan yang dapat disisipkan.

#### DAFTAR PUSTAKA

- [1] Alam, I., (2008). “Aplikasi Kode Huffman dalam Kompresi Gambar Berformat JPEG”, <http://www.informatika.org/~rinaldi/Matdis/2007/2008/Makalah/MakalahIF2153-0708-075.pdf>.
- [2] Filler, T.,(2007). “Imperfect Stegosystems Asymptotic Laws And Near-Optimal Practical Constructions”, Ms, Czech Technical University.
- [3] Fridrich, J., Goljan, M., Hoge, D., (2003). “Steganalysis of JPEG Images: Breaking the F5 Algorithm”. Lecture notes in computer science, vol 2578. Springer, Berlin Heidelberg New York, PP 310 – 323.
- [4] Guillermito, (2002). “Easily Breaking a very weak Steganography Software”, <http://www.guillermito2.net/stegano/camouflage/index.html>.
- [5] Leung, (2004). “Image Compression Standards”. [http://www2.it.lut.fi/kurssit/06-07/Ti5312400/Materiaali/Paiva\\_2/2-2-3.pdf](http://www2.it.lut.fi/kurssit/06-07/Ti5312400/Materiaali/Paiva_2/2-2-3.pdf).

- 
- [6] Munir, R., (2004), "Pengolahan Citra Digital dengan Pendekatan Algoritmik". Bandung: Informatika.
  - [7] Sellars, D., (1996). "An Introduction to Steganography", <http://www.Cs.Uct.ac.za/courses/cs 400 w/NIS/ paper 99/ d sellars/ stego.htm>.
  - [8] Setiana, 2006. "Steganografi pada File Citra Bitmap 24 Bit untuk Pengamanan data menggunakan metode *Least Significant Bit(LSB) Insertion*".
  - [9] Suhono, Supangkat, H., Juanda, K., (2000). "Watermarking Sebagai Teknik Penyembunyian Hak Cipta Pada Data Digital". Jurnal Departemen Teknik Elektro, Institut Teknologi Bandung.
  - [10] Sutoyo, T., Mulyanto, E., et al., (2009). "Teori Pengolahan Citra Digital", Penerbit Andi, Yogyakarta.
  - [11] Vivien, C., (2008). "Pseudo-Random Number Generators", Part of the Postgraduate Journal Club Series, Mathematics, UQ.
  - [12] Westfield, A., (2001). "F5-steganographic algorithm: High capacity despite better steganalysis", Lecture Notes in Computer Science, vol. 2137, pp. 289-302, Springer Verlag.

## INDEKS PENULIS

### A

Ahmed, Bashar., Ma ti, M.L. Ben., Al Mohajir,  
Ahmad Althunibat : 104  
Aldiyingtyas Feryani, Tito Pinandita, Harjono  
: 117  
Alit Prajatama, Muhammad Rusli, Ni Wayan  
Deriani : 133  
Almatsier, Sunita : 118

### B

Badreddine : 93  
Bahrami, Mahdi., Ghorbani, Mazaher.,  
Arabzad, S. Mohammad : 93

### D

Dahria, Muhammad : 124  
Desiani, Anita; Muhammad Arhami : 128  
D. Ariyus : 140  
Dewi, Ketut Rini A., Wibowo, Januar., Amelia,  
Tan : 93  
Dolly., Pruthi, Amit : 93  
Dwi Saputro Andika : 117

### F

Febria Sri Handayani : 108  
Filler, T : 165  
F. Wang, Y. Zhang dan T. Cao : 141  
Fransisca Losavio, dkk : 104  
Fridrich, J., Goljan, M., Hoge, D : 165

### G

G. Bhatnager, S. Metha & S. Mitra : 133  
Gregor Panovski : 105

### H

Hameed, A., Oudah, Ali : 94  
Handoko, Bambang., Rispianda : 93  
Harfoushi, Osama K : 93  
Haryadi Sarjono dan Winda Julianita : 104, 107  
Hidayat, Rachmat : 93

### I

I. F. Elashry, O. S. Farag Allah dan A. M.  
Abbas : 143  
I Gusti Made Trisnayoga, Muhammad Rusli, I  
Wayan Ardi Yasa : 133  
I Padayachee, O. Kotze, and A. Van Der Merwe  
: 103

Indah, Meilia Nur, et.al : 125  
Imam Hambali, M.J. Dewiyani S, Teguh  
Sutanto : 133

### K

Khaligh, A.A., Miremadi, A., Aminilari, M : 93  
Kristanto, Andri : 117

### L

Li, H., Hong, J : 93

### M

M. Al'Amin, Musreny Alda : 133  
M. Mishra dan H. V. Mankar : 141  
Made Purwantara, Suriyati, Zaenal Abidin : 147  
Muhammad Arsyad : 156, 157  
Muhammad Ridho Ardiansyah : 133

### O

Olupot, Charles., Kituyi, Mayoka G., Noguera,  
Jose : 92,93

### P

P. J. Rani dan S. D. Bhavani : 142  
PP Republik Indonesia Nomor 37 : 146

### R

R. Munir : 141, 142, 172  
R. Munir, B. R. dan S. S : 142  
R. Sadikin : 142  
Ramsden, P. : 147  
Ranida, P., Nurul, H : 148  
Rifki Indra Perwira, Anifudin Aziz : 156  
Riza. N., Thamrin, Siregar : 125  
Roger S. Pressman : 103

### S

Sommerville, Ian : 94  
Shelly, G. B., Rosenblatt, H. J : 94  
Shalahuddin, Muhammad dan Rosa Ariani S :  
117, 118  
Siang, Jong Jek : 131  
Sri Kusumadewi, : 128  
Sri Rahayu : 157  
Sudaryono : 124  
Sutojo T., dkk : 125, 127

## **T**

Thamer A. Alrawashdeh, Mohammad Muhairat,  
and Tjandra Yoga Aditama : 157

Tonny Hidayat : 133

Turban, Efraim., Aronson, Jay E., dan Liang,  
Ting-Peng : 156

## **U**

U. M. Bokhari, S. Alam dan S. F. Masoodi : 142

UURI Nomor 14 : 146

## **W**

Wahab, Samsudin : 93

Westfield, A., : 165