

# STEGANOGRAFI DENGAN METODE PENGGABUNGAN FILE MELALUI *COMMAND PROMPT* SERTA STEGANALISIS HASIL DENGAN METODE POLA PENGENALAN GAMBAR, KULTUR GAMBAR RGB 24 BIT DAN RENTANG UKURAN PADA FILE JPEG

Yunita Sartika Sari<sup>1</sup>, Nia Rahma Kurnianda<sup>2</sup>, Dewi Kusumaningsih<sup>3</sup>

Program Studi Magister Ilmu Komputer, Universitas Budi Luhur

Jl. Ciledug Raya, Pertukangan Utara, Jakarta Selatan, 12260, Indonesia

<sup>1</sup>lovelythatha@gmail.com, <sup>2</sup>nia.kurnianda@gmail.com, <sup>3</sup>dewi.kusumaningsih@gmail.com

## ABSTRAK

*Kerahasiaan pada suatu file bukan hanya untuk alasan keamanan, namun juga menyamarkannya dari penglihatan atau menyembunyikan secara rapi agar tidak menimbulkan kecurigaan bagi kasat mata yang memandangnya sehingga file rahasia yang terkandung pada file lain tidak menarik rasa keingin tauhan orang lain yang melihatnya. Pada seni menyembunyikan pesan atau file, metode penggabungan file termasuk salah satu cara sederhana yang dapat menjadi alternatif yang bisa dipilih dan dipergunakan dengan mudah tanpa memerlukan peralatan khusus atau menguasai bahasa pemrograman tertentu karena hanya bermodalkan perintah dari command prompt, dimana pengguna dapat dengan cepat menyembunyikan suatu file di dalam file lain untuk menyamarkannya. Berangkat dari modal tersebut, kami mendapatkan hasil keluaran yang kami analisa dengan menggunakan beberapa metode. Pertama-tama kami menggunakan metode pola pengenalan gambar untuk mengukur perubahan pola pada file asal dan file hasil. Kami juga melakukan pengukuran terhadap perubahan kultur warna antara file asal dan file hasil dengan menggunakan metode pengujian kultur gambar dalam RGB 24 Bit serta pengukuran tingkat kewajaran pada kewajaran ukuran dari file hasil telah kami lakukan dengan melakukan perhitungan rentang ukuran wajar pada file jpg.*

**Kata kunci :** *Steganografi, Command Prompt, RGB 24 Bit, Steganalisis Euclidean*

## I. PENDAHULUAN

Dewasa ini, proses pertukaran data dan informasi termasuk pengiriman pesan dapat dilakukan dalam berbagai macam cara. Selain itu, pesan yang dapat dikirim pun tidak terbatas pada pesan yang berbentuk data teks saja, tetapi bisa juga berbentuk gambar, video, suara dan lain sebagainya.

Media penyampaian pesanpunsemakin berkembang seiring dengan pertumbuhan teknologi di bidang informasi, kita tidak hanya bisa menggunakan jasa pos untuk mengirimkan pesan, tetapi juga bisa menggunakan layanan e-mail yang tersedia melalui internet. Walau pun pengiriman pesan melalui e-mail dapat diterima secara langsung oleh penerima pesan, tetapi pengiriman dengan cara ini juga rentan terhadap bahaya pencurian data.

Atas dasar segala derita dan keragu-raguan atau rasa was was akan keamanan pesan kita tersebut, maka tentunya dengan sebuah metode sederhana dapat kita aplikasikan untuk dapat sedikit mengecoh ancaman tersebut, meski cara ini bukanlah sesuatu yang menjanjikan keamanan tingkat tinggi yang menjamin pesan anda aman.

Berdasarkan permasalahan tersebut diatas,, kami mencoba untuk membuat suatu teknik steganografi sederhana dengan metode penggabungan file melalui command prompt serta steganalisis hasil dengan metode pola pengenalan gambar,

kultur gambar RGB 24 bit dan rentang ukuran pada file JPG. dimana media yang kita perlukan hanyalah berbasis *command prompt* yang ada pada fitur windows. Command prompt pada awalnya dikembangkan oleh Therese Stowell [1]. untuk Windows NT yang kemudian digunakan pada berbagai variasi sistem operasi Windows menggantikan COMMAND.COM dari era MS-DOS dan Windows 9x. Namun untuk sedikit mempermanis tampilan berwarna hitam putih berbasis teks tersebut, kami akan mengaplikasikan metode tersebut dalam user interface berbasis bahasa VB6

## II. LANDASAN TEORI

Pada penulisan ini, penulis membagi dua tahap dari metode yang akan digunakan. Tahap pertama merupakan tahap simulasi dan tahap berikutnya adalah tahap pengujian. Berikut metode yang melandasi kami di setiap tahapannya:

### A. Tahap Simulasi

Pada tahap simulasi, metode yang dipergunakan adalah:

#### 1. Penyembunyian File (Steganografi)

Steganografi berasal dari kata *setganos* dalam bahasa Yunani yang berarti tertutup atau rahasia dan *grahpy* yang berarti tulisan atau gambar. Menurut Eric Cole secara sederhana steganografi adalah tulisan rahasia, bisa dalam

bentuk tinta tak kasat mata pada kertas atau informasi hak cipta yang disembunyikan pada berkas suara, tetapi saat ini steganografi lebih sering dikaitkan dengan versi teknologi tinggi dimana suatu data disembunyikan pada data lain di dalam berkas elektronik[2].

Tujuan dari steganografi adalah merahasiakan atau menyembunyikan keberadaan dari sebuah pesan tersembunyi atau sebuah informasi. Dalam prakteknya, kebanyakan pesan disembunyikan dengan membuat perubahan tipis terhadap data digital lain yang isinya tidak akan menarik perhatian dari penyerang potensial, sebagai contoh sebuah gambar yang terlihat tidak berbahaya.

2. Metode Penggabungan File

Dasar teori yang melandasi dari metode tersebut ialah teknik dengan menggunakan sebuah perintah yang di eksekusi langsung pada command prompt atau cmd.exe.

Berdasarkan pengertian dari Wikipedia cmd.exe adalah penerjemah baris perintah pada Sistem Operasi berbasis Windows CE dan Windows NT (termasuk Windows 2000, Windows XP, Windows Vista, Windows Server 2003 dan Windows Server 2008). Cmd.exe juga merupakan bentuk analog dari COMMAND.COM yang ada pada MS-DOS dan sistem Windows 9x atau dari Unix shell yang digunakan pada sistem *Unix-like*

3. Metode Perancangan Aplikasi

Jika menggunakan cmd.exe maka simulasi yang digunakan berbasis text. Untuk mempermudah bagi yang tidak mudah menghafalkan perintah cmd, kami membuat rancangan aplikasi sederhana untuk melakukan *encode* steganografinya. Namun untuk proses *decode* tidak akan kami buat karena untuk membuka hasil steganografi sederhana ini, cukup dengan menggunakan aplikasi winrar.

Dalam analisa dan perancangan bagi aplikasi steganografi sederhana ini, kami menggunakan konsep perancangan berorientasi obyek dan menggunakan model yang dikenal dengan *unified model Language (UML)* UML dirilis pertama kali pada Oktober 1995 oleh Grady Booch dan James Rumbaugh[3]. Saat ini merupakan sebuah “bahasa” yang telah menjadi standar dalam industri *visualisasi*, merancang dan mendokumentasikan sistem piranti lunak. Adapun model-model yang kami gunakan adalah sebagai berikut:

- a. *Use Case Diagram*
- b. *Class Diagram*
- c. *Sequence Diagram*

**B. Tahap Pengujian**

Dalam tahap pengujian ini, kami menggunakan tiga metode untuk mengukur tingkat efektifitas dari hasil steganografi sederhana ini. Metode-metode tersebut adalah:

1. Pola Pengenalan Gambar

Metode pola pengenalan gambar ini sebenarnya merupakan turunan dari beberapa konsep lain yang diambil dari bidang ilmu statistik, matematika dan fisika. Berikut

adalah konsep-konsep yang menjelaskan cara kerja metode ini:

a. Konsep Simpangan Baku (Standar Deviasi) untuk Mengukur Penyebaran Pola Tekstur Warna

Dalam statistika dan probabilitas, simpangan baku atau deviasi standar adalah ukuran sebaran statistik yang paling lazim. Singkatnya, ia mengukur bagaimana nilai-nilai data tersebar. Bisa juga didefinisikan sebagai, rata-rata jarak penyimpangan titik-titik data diukur dari nilai rata-rata data tersebut.

Simpangan baku didefinisikan sebagai akar kuadrat varians[4]. Simpangan baku merupakan bilangan tak-negatif, dan memiliki satuan yang sama dengan data. Misalnya jika suatu data diukur dalam satuan meter, maka simpangan baku juga diukur dalam meter pula. Untuk bagian pembahasan nanti, rumus simpangan baku atau standar deviasi yang akan digunakan adalah dengan rumus simpangan baku sampel yaitu:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x}_i)^2}$$

b. Konsep Mean untuk Mengukur Rata-Rata dari Pola Tekstur

Rata-rata atau Mean merupakan ukuran statistik kecenderungan terpusat yang paling sering digunakan. Rata-rata ada beberapa macam, yaitu rata-rata hitung (aritmatik), rata-rata geometrik, rata-rata harmonik dan lain-lain. Tetapi jika hanya disebut dengan kata "rata-rata" saja, maka rata-rata yang dimaksud adalah rata-rata hitung (aritmatik).

Penghitungan rata-rata dilakukan dengan menjumlahkan seluruh nilai data suatu kelompok sampel, kemudian dibagi dengan jumlah sampel tersebut. Jadi jika suatu kelompok sampel acak dengan jumlah sampel *n*, maka bisa dihitung rata-rata dari sampel tersebut dengan rumus sebagai berikut

$$\mu = \frac{1}{N} \sum_{i=1}^N X_i$$

c. Konsep Entropi untuk Mengukur Perubahan Pada Pola Gambar

Menurut Hong Cheng entropi adalah ukuran dari ketidakpastian konten informasi[5]. Dalam bagian pembahasan, rumus perhitungan entropi yang dipergunakan adalah sebagai berikut:

$$e = - \sum_{i=1}^n p(x_i) \log P(x_i)$$

d. Konsep Energi untuk Mengukur Banyak Energi yang Digunakan Saat Pembentukan Pola Tekstur Gambar

Dalam fisika, energi adalah properti fisika dari suatu objek, dapat berpindah melalui interaksi fundamental, yang dapat berubah bentuknya namun tidak dapat diciptakan maupun dimusnahkan.

Joule adalah satuan SI untuk energi, diambil dari jumlah yang diberikan pada suatu objek (melalui kerja mekanik) dengan memindahkannya sejauh 1 meter dengan gaya 1 newton. Pada bagian pembahasan, rumus perhitungan energi yang digunakan adalah sebagai berikut

$$E_j = \frac{1}{M \cdot x \cdot N} \sum_{x=1}^M \sum_{y=1}^N [P_j(x, y)]^2$$

e. Konsep Homogeneity untuk Mengukur Keceragaman Pola Tekstur Gambar

Pengujian homogenitas dilakukan untuk mengetahui apakah variansi antara kelompok yang diuji berbeda atau tidak[6]. Pada pembahasan, rumus yang digunakan untuk pengukuran homogeneity tersebut adalah:

$$H = \sum_i \sum_j \frac{P_d(i, j)}{1 + |i - j|}$$

f. Konsep Jarak Euclidean untuk Mengukur Kemiripan Pola Tekstur Gambar

Jarak Euclidean adalah metrika yang paling sering digunakan untuk menghitung kesamaan 2 vektor[7]. Jarak Euclidean antara titik p dan q adalah panjang ruas garis yang menghubungkan mereka. Dalam koordinat Cartesian, jika p = (p<sub>1</sub>, p<sub>2</sub>, ..., p<sub>n</sub>) dan q = (q<sub>1</sub>, q<sub>2</sub>, ..., q<sub>n</sub>) adalah dua poin di Euclidean n-space, maka jarak (d) dari p ke q, atau dari q ke p diberikan oleh rumus Pythagoras. Untuk pembahasan, kita menggunakan rumus jarak Euclidean sebagai berikut:

$$\|a-b\| = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2 + \dots + (a_n - b_n)^2}$$

2. Pengujian Kultur Gambar dengan RGB 24 Bit

MATLAB menyimpan dan menyajikan gambar hitam putih dalam bentuk matrik, dimana setiap elemen matrik mewakili satu piksel pada gambar. Sebagai contoh, sebuah gambar yang terdiri dari M baris dan N kolom titik-titik yang berbeda akan disimpan MATLAB sebagai M X N matrik. Gambar berwarna membutuhkan matrik 3 dimensi, dalam hal ini matrik M X N X3, dimana lapisan pertama mewakili warna merah, lapisan kedua mewakili warna hijau dan lapisan ketiga mewakili warna biru.

Format berkas MATLAB menyimpan warna dalam bentuk gambar 24 bit RGB, dimana komponen warna merah, hijau dan biru masing-masing menggunakan 8 bit. Komponen ketiga warna tersebut untuk setiap piksel disimpan bersama data array 3 dimensi. Sebagai contoh, komponen merah, hijau dan biru dari piksel (10,5) disimpan sebagai RGB (10,5,1), RGB (10,5,2) dan RGB (10,5,3) sesuai urutan[8].

Pada pengujian, gambar akan ditampilkan dalam bentuk matrix yang menunjukkan lapisan RGB.

3. Kewajaran Rentang Ukuran pada File JPEG

Pada pengukuran melalui file size, kami menggunakan dua batas. Yaitu kemungkinan batas atas dan batas bawah dengan metode pengukuran file size jpg konversi dari bmp maupun dengan menggunakan standar photoshop dengan konversi bmp ke jpg.

Dalam metode pengukuran tersebut, pertama-tama kami hitung nilai byte dari versi bmp nya. Dengan menggunakan rumus sebagai berikut:

$$\text{Bmp Size} = (\text{width} \times \text{height} \times \text{bit depth}) / 8$$



Jika kita penggunaan metode standar konversi bmp ke jpg, maka hasil perhitungan file bmp tersebut kita kali dengan 10%, maka didapatkan hasil size JPEG nya[9].

Jika menggunakan standar konversi bmp ke jpg dengan standar photoshop, maka pendekatan perhitungannya terbagi atas tiga jenis.

- a) Baik (1/4 size bmp)
- b) Standar (1/8 size bmp)
- c) Dasar (1/16 size bmp)

C. Bahan-Bahan

Sebagai tahap persiapan, penulis memilih 1 buah gambar animasi yang penulis pinjam dari google.com dan sebuah file dalam format .ppt yang telah dikompresi sebelumnya ke dalam ekstensi .rar. Berikut adalah capture dari gambar dan file yang akan dipergunakan dalam proses simulasi dengan rincian sebagai berikut:

No	File	Nama File	Format	Alamat Penyimpanan
1.		01	jpg	Drive D:\New folder\coba\01.jpg
2.		DW_Eksekutif	rar	Drive D:\New folder\coba.rar

Gambar 1: File yang Akan dipergunakan untuk Simulasi

Profil gambar *cover* yang kami gunakan adalah sebagai berikut:

Type : JPEG  
 Width : 259 pixels  
 Height : 184 pixels  
 Bit Depth : 24 (RGB)  
 Size : 8,41 Kb  
 Size On Disk : 12 Kb

Profil file yang akan disisipkan yang dipergunakan adalah sebagai berikut:

Type : RAR  
 Size : 32,8 Kb  
 Size On Disk : 36 Kb

Kemudian, yang kita butuhkan berikutnya adalah:

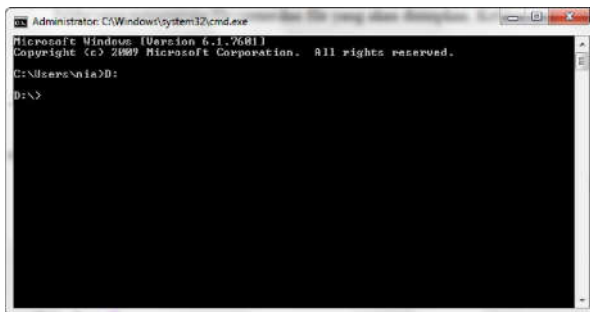
- 1) *Command Prompt* (cmd.exe) untuk uji coba secara manual pada konsep yang diterapkan
- 2) *Frame work Visual Basic 6.0* untuk pembuatan aplikasi user interface.
- 3) Aplikasi WinRAR untuk Membuka Hasil Steganografi
- 4) Aplikasi Matlab 7.1 untuk Steganalisis

### III. PEMBAHASAN

#### A. Simulasi Konsep Secara Manual dengan *Command Prompt*

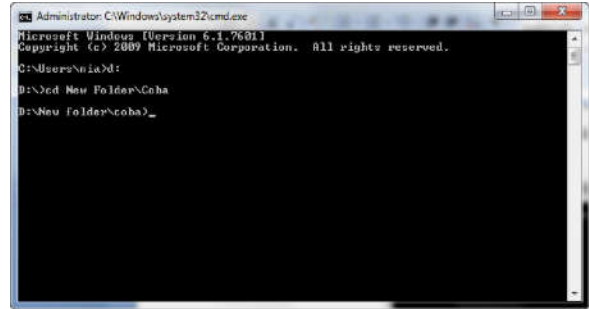
Berikut langkah-langkah yang dipergunakan pada cmd.exe:

1. Satukan *file Cover* dan file yang akan disisipkan ke dalam satu folder. Dalam simulasi ini, kami meletakkan kedua file tersebut ke dalam drive D:\New Folder\coba
2. Ketika awal kami melakukan running cmd.exe, maka default folder yang akan kita kunjungi adalah folder user yang sedang aktif di drive C:\ maka kita perlu memasuki folder yang menampung *file Cover* dan file yang akan disisipkan. Ketikkan perintah D: kemudian tekan enter untuk memasuki Drive yang dituju



Gambar 2: Memasuki Drive D:\

3. Kemudian masuki folder ke tempat data disimpan dengan mengetikkan perintah `cd New Folder\Coba` kemudian tekan enter untuk memasuki folder yang dituju

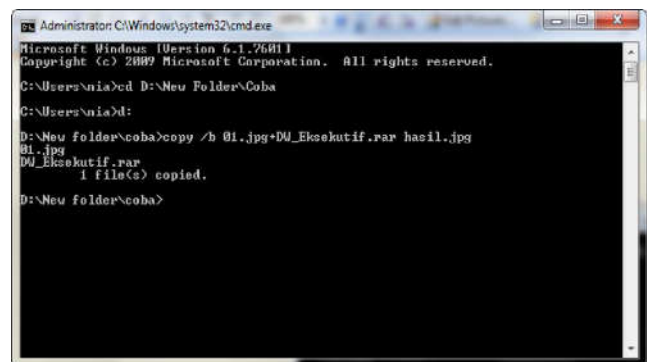


Gambar 3: Memasuki Folder Coba

4. Setelah masuk ke dalam folder tersebut, ketikkan perintah:

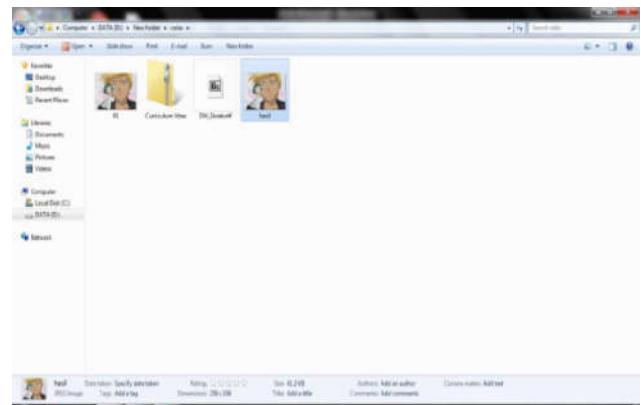
```
copy /b 01.jpg+DW_Eksekutif.rar hasil.jpg
```

Kemudian akan terdapat hasil seperti dibawah ini:



Gambar 4: Selesai Copy

5. Kemudian buka eksplorer anda menuju Drive D:\New Folder\coba. Jika perintah tadi berhasil dilakukan, maka akan terdapat file *hasil.jpg*.

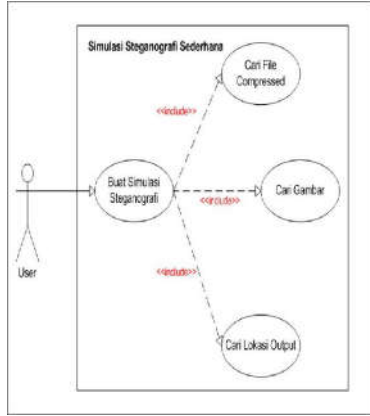


Gambar 5: File hasil.jpg pada folder D:\New Folder\Coba

#### B. Perancangan Aplikasi *User Interface*

Telah penulis paparkan sebelumnya bahwa penulis akan membuat sebuah *user interface* yang akan memudahkan pengguna untuk dapat melakukan *encoding* steganografi dengan metode penggabungan file tersebut. Berikut adalah rancangan dari aplikasi *user interface* tersebut:

1. Use Case



Gambar 6: Use Case Diagram

2. Deskripsi Use Case

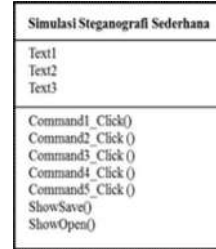
Nama Use Case : Buat Simulasi Steganografi

Actor : User

Deskripsi :

- User membuka program Simulasi Steganografi Sederhana
- Kemudian User mencari File yang telah dikompresi yang memiliki ekstensi .rar dengan menekan *command2* (Button Cari File Compressed). Sistem akan memanggil file explorer secara otomatis
- User memilih file yang akan disisipkan kemudian menekan *button open*. Maka pada *Text1* akan terisi lokasi file yang akan disisipkan
- Setelah itu User mencari File Cover yang berekstensi .jpg dengan menekan *Command3* (Button Cari Gambar). Kemudian sistem secara otomatis akan kembali memanggil file explorer
- User memilih file yang akan dijadikan cover kemudian menekan *button open*. Maka pada *Text2* akan terisi lokasi file yang akan dijadikan cover
- Selanjutnya, user memberi nama dan memilih lokasi penyimpanan hasil dari kompilasi kedua file kemudian menekan *Command4* (Button Cari Lokasi Output). Program akan otomatis memanggil explorer untuk *saving selection*.
- User menekan *button save* untuk menentukan lokasi dan nama file. Maka pada *Text3* akan terisi lokasi dan nama file hasil output
- Untuk memproses kedua file terpilih menjadi output, User menekan *Command1* (Button Proses Sekarang)
- Ketika kedua file telah selesai diproses, maka akan muncul *Message Box* yang menginformasikan bahwa proses telah selesai dengan pesan "Sudah Selesai". User menekan *button Ok* untuk kembali ke form utama.
- User dapat keluar dari form utama dengan menekan *Command5* (Button Batal) maka program akan keluar dengan sendirinya.

3. Class Diagram



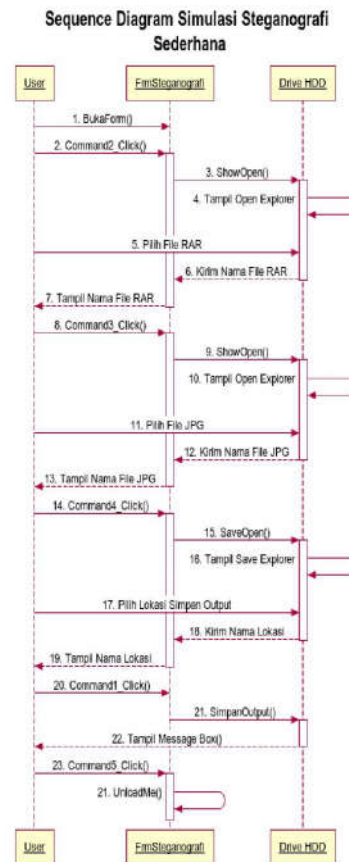
Gambar 7: Class Diagram

4. Rancangan Layar



Gambar 8: Rancangan Layar

5. Sequence Diagram

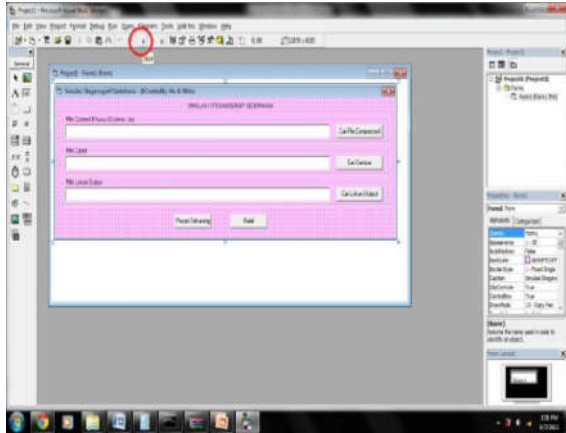


Gambar 9: Sequence Diagram

### C. Simulasi Konsep dengan *Prototype* Aplikasi *User Interface*

Untuk menjalankan *prototype* dari aplikasi yang kami buat tersebut, kami menggunakan langkah-langkah sebagai berikut:

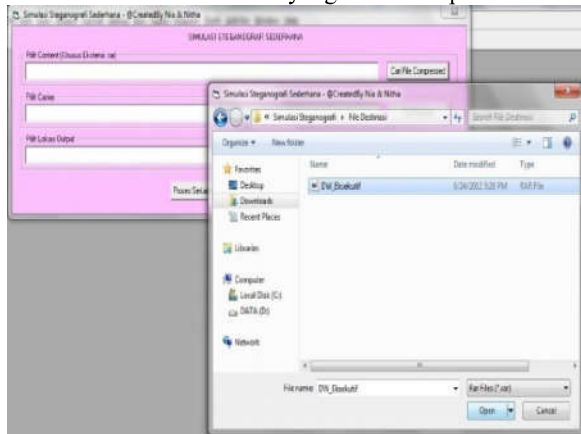
- 1) *Run* program dengan menekan *button start* pada *toolbar*. Pada gambar dibawah ini, tombol tersebut diberi bulatan merah:



Gambar 10: Run Program

- 2) Setelah program berhasil di *running*, maka selanjutnya adalah mencoba menggunakan program tersebut.

Pertama-tama kita akan mencari file yang akan disisipkan dengan cara menekan *button cari file compressed*. Maka otomatis program akan memanggil file *explorer* yang memiliki default *drive C:\* dan mencari file yang telah dikompresi dengan ekstensi *.rar*. Anda dapat menjelajah sesuka hati untuk mencari file yang akan disisipkan.

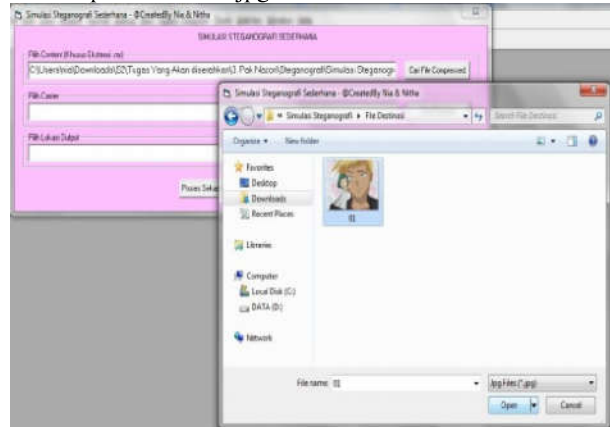


Gambar 11: Cari File Compressed

Jika telah selesai mencari file yang akan sisipkan tersebut pada *explorer*, maka tekan *button open* sehingga tampilan kembali kepada form utama dan *textbox* disebelah *button cari file compressed* akan terisi dengan alamat lokasi file berekstensi *.rar* tersebut yang telah dipilih tadi.

Kemudian setelah kita memilih file yang akan disisipkan, kita memerlukan file lain untuk menutupi file tersebut. Maka kita dapat mencarinya dengan menekan *button cari gambar*.

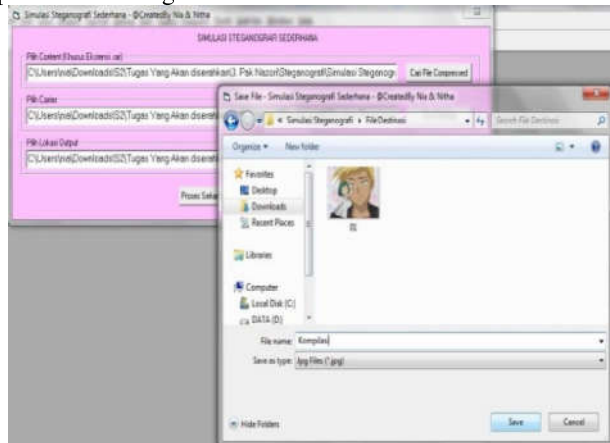
Maka sistem akan membuka kembali *open explorer* dengan berfokus pada ekstensi *.jpg*



Gambar 12: Cari Gambar

Jika kita telah selesai mencari *file* yang akan kita jadikan *cover* tersebut pada *explorer*, maka kita tekan *button open* sehingga tampilan kembali kepada form utama dan *textbox* disebelah *button cari gambar* akan terisi dengan alamat lokasi file berekstensi *.jpg* tersebut yang telah dipilih tadi.

Kemudian setelah file yang akan menyembunyikan dan disembunyikan terpilih, maka cari lokasi untuk menyimpan dengan cara menekan tombol cari lokasi output. Saat *explorer* terbuka, maka cari lokasinya dan isi namanya. Pada simulasi ini menamakan pada file hasil dengan nama *kompilasi.jpg* pada kotak dialog kemudian tekan *save*.



Gambar 13: Cari Lokasi Simpan

Setelah tekan proses sekarang dan buka, akan terlihat oleh anda file dengan nama *kompilasi.jpg*

### D. Hasil Simulasi Konsep

Pada simulasi ini, metode yang digunakan adalah sama baik secara manual ataupun dengan menggunakan aplikasi. Hasil keluaran yang didapat adalah sama meskipun ada sedikit perbedaan pada mekanismenya. Hasil keluaran adalah file dalam format *JPEG*. Format dan gambar yang dikenali oleh

explorer adalah sama dengan gambar cover yang telah kita pilih (01.jpg) namun ketika kita membukannya dengan bantuan aplikasi WinRAR, maka baru akan terlihat bahwa sebenarnya ada file DW\_Eksekutif.rar terkandung di dalamnya.

Pada hasil keluaran tersebut jika kita klik kanan dan memilih properties, maka akan tetap pada bagian type file akan dikenali berekstensi .jpg sehingga jika membukanya langsung dengan klik dua kali, file tersebut akan terbuka dengan menggunakan Ms. Office Picture Manager atau dengan Windows Photo Viewer.

Untuk profil ukuran file, konsepnya adalah file cover (01.jpg) ditambah dengan file yang disisipkan (DW\_Eksekutif.rar) sehingga meskipun dalam format.jpg namun terjadi peningkatan ukuran file.

Tabel 1: Tabel Properties

No.	Keterangan	Sebelum	Sesudah
1.	Format	JPEG	JPEG
2.	Ukuran	12 Kb	44 Kb
3.	Dimensi	194x259	194x249
4.	Bit Depth	24	24

#### IV. PENGUJIAN DAN ANALISA HASIL

##### A. Steganalisis dengan Metode Pola Pengenalan Gambar

Dalam pengukuran hasil simulasi ini, kami menggunakan bantuan program matlab 7.1. berikut penulis sampaikan listing kode program yang di running dengan matlab 7.1 untuk menguji tekstur gambar:

```
% membersihkan layar untuk menampung hasil tampilan lebih banyak
clear, close all
clc
```

```
% deklarasi gambar asal
imggambar1 = imread('D:\New folder\coba\01.jpg');
```

```
% deklarasi gambar hasil
imgtester = imread('D:\New folder\coba\kompilasi.jpg');
```

```
% Pola tekstur gambar 01.jpg Sebelum Disisipkan File DW_Eksekutif.rar
```

```
std_1 = std2(imggambar1);
avg_1 = mean2(imggambar1);
ent_1 = entropy(imggambar1);
eng1 = graycoprops(imggambar1,{'energy'});
eng_1 = struct2array(eng1);
hgi1 = graycoprops(imggambar1,{'homogeneity'});
hgi_1 = struct2array(hgi1);
CITRAGAMBAR1 = [std_1, avg_1, ent_1, eng_1, hgi_1]
```

```
% Pola Tekstur Gambar kompilasi.jpg (gambar 01.jpg Setelah disisipkan File DW_Eksekutif.rar)
```

```
std_10 = std2(imgtester);
avg_10 = mean2(imgtester);
ent_10 = entropy(imgtester);
eng10 = graycoprops(imgtester,{'energy'});
eng_10 = struct2array(eng10);
```

```
hgi10 = graycoprops(imgtester,{'homogeneity'});
hgi_10 = struct2array(hgi10);
CITRATESTER = [std_10, avg_10, ent_10, eng_10, hgi_10]
```

```
% Euclidean Gambar 01 Terhadap Hasil
```

```
SOURCE_GAMBAR1=sqrt((CITRAGAMBAR1(1,1)-CITRATESTER(1,1))^2+(CITRAGAMBAR1(2,1)-CITRATESTER(2,1))^2+(CITRAGAMBAR1(3,1)-CITRATESTER(3,1))^2+(CITRAGAMBAR1(4,1)-CITRATESTER(4,1))^2+(CITRAGAMBAR1(5,1)-CITRATESTER(5,1))^2)
```

```
% Euclidean Gambar Hasil Terhadap Hasil
```

```
SOURCE_TESTER=sqrt((CITRATESTER(1,1)-CITRATESTER(1,1))^2+(CITRATESTER(2,1)-CITRATESTER(2,1))^2+(CITRATESTER(3,1)-CITRATESTER(3,1))^2+(CITRATESTER(4,1)-CITRATESTER(4,1))^2+(CITRATESTER(5,1)-CITRATESTER(5,1))^2)
```

%Berikut hasil dari running pengujian kedua gambar tersebut:

```
CITRAGAMBAR1 =
```

```
59.7511
165.6723
7.3598
0.0000
0.0000
0.0361
0.0359
0.0357
```

```
CITRATESTER =
```

```
59.7511
165.6723
7.3598
0.0000
0.0000
0.0361
0.0359
0.0357
```

```
SOURCE_GAMBAR1 = 0
```

```
SOURCE_TESTER = 0
```

Kami transformasikan hasil pengukuran diatas ke dalam bentuk tabel agar mudah terlihat hasilnya seperti berikut:

Tabel 2: Hasil Steganalisis Pola Pengenalan Gambar

No	Aspek Perbandingan	Gambar 01.jpg	Gambar Hasil.jpg
1	Kewajaran Penyebaran Pola Tekstur Warna	59.7511	59.7511
2	Rata-Rata Pola Tekstur	165.6723	165.6723
3	Entropi	7.3598	7.3598
4	Tekstur Keabuan	0	0
5	Struktur Tekstur Keabuan	0	0
6	Homogeinity	0.0361	0.0361
7	Struktur Tekstur Homogeinity	0.0359	0.0359
8	Kemiripan Tekstur (Jarak Euclidean)	0	0

Dari tabel tersebut, kita dapat simpulkan bahwa Kedua gambar mirip satu sama lain yang artinya tidak ada perubahan antara sebelum dan sesudah di sisipkan file lain di dalamnya

**B. Steganalisis dengan Metode Pengujian Kultur Gambar dalam RGB 24 Bit**

Untuk pengukuran kultur gambar, kami menggunakan tes kultur warna dalam bentuk RGB. Tes tersebut juga menggunakan bantuan aplikasi Matlab 7.1. berikut kami sampaikan script Matlab yang kami gunakan dalam mengukur apakah kultur warna pada gambar terdapat perubahan atau tidak. Namun untuk hasil akan kami sampaikan dalam bentuk pengambilan *sample* dikarenakan cukup panjang. Untuk mencobanya, kami menyarankan untuk mengkompilasi satu persatu dari sript dibawah ini:

- 1) Kompilasi pertama : Konversi gambar 01 ke dalam kultur RGB
 

```
% membersihkan layar untuk menampung hasil tampilan lebih banyak
clear, close all
clc
% deklarasi gambar 01
imggambar1 = imread(' D:\New folder\coba\01.jpg');
% konversi gambar 01 menjadi bentuk RGB
ascii=uint8(imggambar1)
```
- 2) Kompilasi Kedua : Konversi gambar kompilasi ke dalam kultur RGB
 

```
% membersihkan layar untuk menampung hasil tampilan lebih banyak
clear, close all
clc
% deklarasi gambar kompilasi
Imggambar2 = imread(' D:\New folder\coba\kompilasi.jpg');
% konversi gambar kompilasi menjadi bentuk RGB
ascii=uint8(imggambar2)
```

Pada hasil perbandingan dari kedua gambar pada kultur RGB nya adalah serupa, sehingga tidak ditemukan perubahan pada gambar Cover setelah disisipkan file lain didalamnya. Kami mengambil beberapa sample dari RGB hasil pengujian sebagai berikut:

Tabel 3: Hasil Steganalisis Kultur Gambar

No	Kolom / Baris	Kultur R/G/B 01.jpg	Kultur R/G/B Kompilasi.jpg
1	1 / 5	221/222/234	221/222/234
2	10/10	225/222/233	225/222/233
3	15/17	227/220/229	227/220/229
4	19/6	229/214/227	229/214/227
5	25/21	231/223/234	231/223/234

**C. Steganalisis dengan Rentang Ukuran File JPEG**

Pada pengukuran melalui *file size*, kami menggunakan dua batas. Yaitu kemungkinan batas atas dan batas bawah dengan metode pengukuran *file size* jpg konversi dari bmp maupun

dengan menggunakan standar photosop dengan konversi jpg-bmp. Profil gambar carrier yang kami pergunakan adalah sebagai berikut:

Width : 259 pixels  
 Height : 184 pixels  
 Bit Depth : 24 (RGB)

Dalam metode pengukuran tersebut, pertama-tama kami hitung nilai byte dari versi bmp nya. Dengan menggunakan rumus sebagai berikut:

$$\text{Bmp Size} = (\text{width} \times \text{height} \times \text{bit depth}) / 8$$

Sehingga besar file bmpnya menjadi seperti berikut:  $(259 \times 194 \times 24) / 8 = 150.738$  (150 Kb)

Jika kita pergunakan metode standar konversi bmp ke jpg, maka hasil perhitungan file bmp tersebut kita kali kan dengan 10%:  $10\% \times 150.738 = 15,0738$  (15 Kb)

Jika menggunakan standar konversi bmp ke jpg dengan standar photoshop, maka pendekatan perhitungannya terbagi atas tiga jenis.

- a. Baik (1/4 size bmp)
- b. Standar (1/8 size bmp)
- c. Dasar (1/16 size bmp)

Maka kita masukan nilai tersebut kepada ketiga jenis tersebut:

- a. Jpg ukuran terbaik =  $(1/4) \times 150.738 = 37.648,5$  (38 Kb)
- b. Jpg ukuran sedang =  $(1/8) \times 150.738 = 18.842,25$  (19 Kb)
- c. Jpg ukuran rendah =  $(1/16) \times 150.738 = 9.421,125$  (9,4 Kb)

Maka kita dapatkan rentang ukuran file gambar dengan dimensi 259x194 dalam ekstensi jpg adalah 9,4 Kb s/d 38 Kb dianggap sebagai size wajar dari gambar dan rata-rata besarnya file jpg tersebut adalah 22,2 Kb namun size sebenarnya akan tergantung dengan banyaknya keragaman warna dari gambar tersebut.

Mari kita bandingkan pada gambar asal dan gambar hasil dengan range size wajar gambar, maka akan kita dapatkan hasil sebagai berikut:

Tabel 4: Hasil Range Gambar

Size Perkiraan	01.jpg (12 Kb)	Hasil.jpg (44 Kb)
15 Kb	Lebih Kecil	Lebih Besar
Range 9,4 s/d 38 Kb	Masuk Range	Diluar Range

Maka, ukuran 12 Kb tersebut pada gambar 01.jpg dikatakan wajar dan pada gambar hasil.jpg sebesar 44 Kb menjadi tidak wajar, ukuran akan semakin besar seiring dengan besarnya ukuran file yang disisipkan.

**D. Kelebihan dan Kekurangan**

- 1) Kelebihan
 

Kelebihan dari teknik steganografi sederhana dengan metode penggabungan file yaitu :

  - a) Proses dalam menggabungkan file relative cepat
  - b) Tidak mengubah pola pada gambar



c) Mudah dalam menggunakan metode ini

2) Kekurangan

Pada teknik steganografi sederhana ini terdapat kekurangan, diantaranya adalah :

- a) Tingkat keamanannya rendah
- b) Mudah dideteksi jika melihat ukuran file yang besar

## V. PENUTUP

### A. Kesimpulan

Setelah melaksanakan simulasi steganografi dengan menggunakan file yang disisipkan ke dalam gambar menggunakan metode penggabungan file melalui *command prompt* dan diadakan pengukuran dari hasilnya, maka dapat disimpulkan:

1. Gambar carrier tidak mengalami perubahan tekstur dan kultur pada gambar setelah disisipkan file.
2. Mendeteksi kewajaran file salah satunya dapat dilakukan dengan metode pengukuran file size jpg. File gambar yang disisipkan file berada diluar batas kewajaran ukuran file karena merupakan ukuran gabungan dari dua file
3. Hasil yang dilakukan baik secara manual melalui cmd ataupun aplikasi adalah sama karena menggunakan metode yang sama

### B. Rekomendasi

Setelah melaksanakan simulasi tersebut, kami menjadi lebih paham akan cara kerja steganografi dalam level yang sangat sederhana.

Untuk pengembangan yang lebih baik, kami berharap dapat menerapkan tambahan metode menggunakan kriptografi

untuk dapat melakukan enkripsi file yang akan disisipkan untuk mendukung keamanan data yang lebih baik.

## DAFTAR PUSTAKA

- [1] Zachary, G. Pascal (1994). *Showstopper! The Breakneck Race to Create Windows NT and the Next Generation at Microsoft*, New York: The Free Press.
- [2] Cole, Eric (2003). *Hiding in Plain Sight: Steganography and the Art of Covert Communication*, Indianapolis: Wiley Publishing, Inc.
- [3] Booch, G., Rumbaugh J., Jacobson I. (1998). *The Unified Modeling Language user Guide*, Reading: Addison Wesley.
- [4] Supranto, J. (2000). *Statistik Teori dan Aplikasi Ed. 6*, Jakarta: Erlangga.
- [5] Cheng, Hong (2015) *Sparse Representation, Modeling and Learning in Visual Recognition: Theory, Algorithms and Applications*, London: Springer-Verlag London Ltd.
- [6] Nisfiannoor, Muhammad (2009). *Pendekatan Statistika Modern untuk Ilmu Sosial*, Jakarta: Penerbit Salemba Humanika.
- [7] Putra, Darma (2010). *Pengolahan Citra Digital*, Yogyakarta: Penerbit ANDI.
- [8] Gan, W., M. Kuo, S. (2007). *Embedded Signal Processing with the Micro Signal Architecture*, Hoboken: John Wiley and Son, Inc.
- [9] Wijanarko, Lizard (2010). Cara Mengitung Ukuran File Gambar dan Foto Digital. url: [http://www.AhliDesain.com/cara\\_menghitung\\_ukuran\\_file\\_gambar\\_dan\\_foto\\_digital.html](http://www.AhliDesain.com/cara_menghitung_ukuran_file_gambar_dan_foto_digital.html). diakses pada 10 Juni 2015.