

# Analisis Penerapan Metode Lossy Pada Kompresi Citra Steganografi

Abdul Sani Sembiring

STMIK Budi Darma Medan, Jl. SM.Raja No.338 Sp.Limun Medan, Sumut, Indonesia

[www.stmik-budidarma.ac.id](http://www.stmik-budidarma.ac.id) // E-mail : [gurkiy@gmail.com](mailto:gurkiy@gmail.com)

## ABSTRACT

The internet has become a very important part of communication development in the world. Exchange of information over the internet has many advantages compared to other communication media, especially in terms of speed. But information sent via the internet, security cannot be accounted for. Tapping on confidential information, very often occurs in this communication media. Channels used by the internet are generally not safe channels.

By using the lossy compression method, compression can only be applied to images that have similarities to themselves (self similarity) so that the size of the image gets smaller and saves storage space and accelerates the sending of the image using internet media.

**Kata kunci: Lossy, Citra, Kompresi, Steganografi, BPCS**

## PENDAHULUAN

Kompresi citra adalah aplikasi kompresi data yang dilakukan terhadap citra *digital* dengan tujuan untuk mengurangi *redundansi* dari data-data yang terdapat dalam citra sehingga dapat disimpan dan ditransmisikan secara efisien. Pada pemampatan dengan metode Lossy hanya bisa diterapkan untuk citra yang memiliki kemiripan dengan dirinya sendiri (self similarity).

Steganografi adalah suatu seni kuno menyampaikan pesan dengan cara rahasia, sehingga hanya penerima yang mengetahui. Steganografi membutuhkan dua properti, yaitu media penampung dan data rahasia yang akan disembunyikan.

Citra menjadi objek kompresi dan steganografi oleh penulis, dikarenakan kompresi dan steganografi sama-sama dapat menggunakan objek citra yang diangkat penulis. Walaupun sebenarnya banyak objek lain selain citra yang dapat diterapkan pada kompresi dan steganografi.

## LANDASAN TEORI

### 2.1. Kompresi

Kompresi citra adalah aplikasi kompresi data yang dilakukan terhadap citra *digital* dengan tujuan untuk mengurangi *redundansi* dari data-data yang terdapat dalam citra sehingga dapat disimpan atau ditransmisikan secara efisien.

Kompresi citra memiliki manfaat yang besar dalam industri multimedia saat ini. Berikut manfaat yang di dapat kompresi adalah:

1. Pengiriman data (*data transmisson*) pada saluran komunikasi data

Citra yang telah dikompres dapat dikirim dengan waktu yang lebih singkat dibandingkan dengan citra yang tidak dikompres. Contoh aplikasi gambar lewat fax, pengiriman data medis, pengiriman gambar via telepon genggam, *download* gambar dari internet, dan sebagainya.

2. Penyimpanan data (*data storing*) di dalam media sekunder (*storage*)

Citra yang telah dikompres membutuhkan ruang memori di dalam media *storage* yang lebih sedikit dibandingkan basis data gambar, *office automation*, *vidieo storage* (seperti *vidieo compact disc*) dan lain-lain.

Kompresi citra bertujuan untuk meminimalkan jumlah *bit* yang diperlukan untuk merepresentasikan citra. Apabila sebuah foto berwarna berukuran 3 inci x 4 inci diubah ke bentuk digital dengan tingkat resolusi sebesar 500 *dot per inch* (*dpi*), maka diperlukan  $3 \times 4 \times 500 \times 500 = 3.000.000$  dot (*pixel*). Setiap *pixel* terdiri dari 3 *byte* dimana masing-masing *byte* merepresentasikan warna merah, hijau, dan biru. sehingga citra digital tersebut memerlukan *volume* penyimpanan sebesar  $3.000.000 \times 3 \text{ byte} + 1080 = 9.001.080$  *byte* setelah ditambahkan jumlah *byte* yang diperlukan untuk menyimpan *format* (*header*) citra.

Kompresi citra dikembangkan untuk memudahkan penyimpanan dan pengiriman citra. Teknik kompresi yang ada sekarang memungkinkan citra dikompresi sehingga ukurannya menjadi jauh lebih kecil daripada

ukuran asli. Ada dua tipe utama kompresi data, yaitu kompresi tipe *lossless* dan kompresi tipe *lossy*. Kompresi tipe *lossy* adalah kompresi dimana terdapat data yang hilang selama proses kompresi. Akibatnya kualitas data yang dihasilkan jauh lebih rendah daripada kualitas data asli. Sementara itu, kompresi tipe *lossless* tidak menghilangkan informasi setelah proses kompresi terjadi, akibatnya kualitas citra hasil kompresi tidak menurun. Namun demikian, rasio kompresi yang digunakan untuk kompresi tipe *lossless* lebih kecil daripada rasio kompresi pada kompresi tipe *lossy*.

Parameter-parameter citra yang penting dalam proses kompresi diantaranya adalah sebagai berikut

1. Resolusi

Resolusi citra menyatakan ukuran panjang kali lebar dari sebuah citra. Resolusi citra biasanya dinyatakan dalam satuan *pixel*. Semakin tinggi resolusi sebuah citra, semakin baik kualitas citra tersebut. Namun, tingginya resolusi menyebabkan semakin banyaknya jumlah *bit* yang diperlukan untuk menyimpan dan mentransmisikan data citra tersebut.

2. Kedalaman *Bit*

Kedalaman *bit* menyatakan jumlah *bit* yang diperlukan untuk merepresentasikan tiap *pixel* citra pada sebuah *frame*. Kedalaman *bit* biasanya dinyatakan dalam satuan *bit/pixel*. Semakin banyak jumlah *bit* yang digunakan untuk merepresentasikan sebuah citra, maka semakin baik kualitas citra tersebut.

3. Konsep Redundansi

Redundansi merupakan suatu keadaan dimana representasi suatu elemen data tidak bernilai signifikan dalam merepresentasikan keseluruhan data. Keadaan ini menyebabkan data keseluruhan dapat direpresentasikan secara lebih kompak dengan cara menghilangkan representasi dari sebuah elemen data yang redundan. Redundansi yang terdapat pada citra statik adalah redundansi spasial.

Metode kompresi citra berdasarkan *redundansi spasial* diantaranya adalah sebagai berikut :

a. *Subsampling*

*Subsampling* merupakan metode kompresi dengan mengurangi jumlah *pixel* yang diperlukan untuk merepresentasikan suatu citra. *Subsampling* dapat dilakukan dengan dua cara. Cara pertama adalah mengambil *pixel - pixel* tertentu dari citra, misal *pixel - pixel* pada baris dan kolom saja. Cara kedua adalah

dengan mengambil rata-rata dari kelompok *pixel* dan menggunakan nilai tersebut sebagai ganti nilai kelompok *pixel* ini. Cara ini lebih kompleks, tetapi menghasilkan kualitas yang lebih baik. *Subsampling* sebanding dengan pengurangan resolusi

b. Pengurangan kedalaman *bit*

Metode ini dilakukan dengan mengurangi jumlah *bit* yang digunakan untuk merepresentasikan suatu *pixel*. Misalnya dengan mengurangi kedalaman *bit* dari 16 *bit/pixel* menjadi 8 *bit/pixel*. Metode ini mengurangi kualitas citra

c. *Transformation Coding*

*Transformation coding* merupakan transformasi data dari domain ruang ke domain frekuensi. Cara ini menghasilkan data yang lebih mudah diproses untuk kompresi lebih lanjut. Transformasi yang populer digunakan antara lain *Discrete Cosine Transform (DCT)* yang diadopsi dalam standar kompresi *JPEG* dan *Discrete Wavelet Transform (DWT)* yang digunakan dalam kompresi *JPEG 2000*.

Kriteria yang digunakan untuk mengukur pemampatan citra adalah :

1. Waktu kompresi dan waktu dekompresi

Proses kompresi merupakan proses mengodekan citra (*encode*) sehingga diperoleh citra dengan representasi kebutuhan memori yang minimum. Proses dekompresi adalah proses untuk menguraikan citra yang dimampatkan untuk dikembalikan lagi (*decoding*) menjadi citra yang tidak mampat.

2. Kebutuhan memori

Metode kompresi yang baik adalah metode kompresi yang mampu mengompresi *file* citra menjadi *file* yang berukuran paling minimal. Biasanya semakin besar persentase pemampatan, semakin kecil kebutuhan memori yang diperlukan sehingga kualitas citra makin berkurang. Dan sebaliknya, semakin kecil persentase citra yang dimampatkan, semakin bagus kualitas hasil pemampatan tersebut.

3. Kualitas Pemampatan

Metode kompresi yang baik adalah metode kompresi yang mampu mengembalikan citra hasil kompresi menjadi citra semula tanpa kehilangan informasi apa pun. Semakin berkualitas hasil pemampatan, semakin besar memori yang dibutuhkan.

4. Format Keluaran

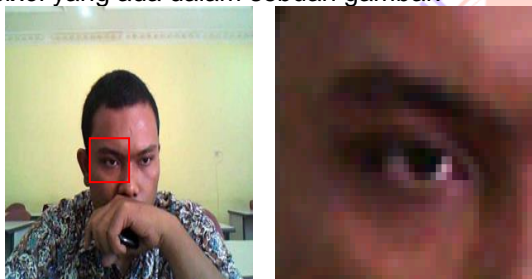
Format citra hasil pemampatan yang baik adalah yang cocok dengan kebutuhan pengiriman dan penyimpanan data.

## 2.2. Citra Digital

### 1. Konsep Citra Digital

Citra digital adalah citra yang dapat diolah oleh komputer. Sebuah citra digital menyimpan data berupa *bit* yang dapat dimengerti oleh manusia dengan visualisasi *bit* tersebut pada kanvas menjadi gambar. Pengolahan yang dapat dilakukan terhadap citra digital antara lain adalah menampilkan bentuk gambar, melakukan perubahan terhadap gambar (*image editing*), dan mencetak citra digital ke atas media berupa kertas.

Citra digital terdiri dari *pixel-pixel* berukuran kecil yang membentuk sebuah bentuk gambar yang dapat dilihat oleh mata manusia. Kepadatan *pixel - pixel* yang ada dalam gambar ini disebut dengan resolusi. Semakin besar resolusi sebuah citra digital maka kualitas gambar dari citra digital tersebut semakin baik. Gambar 1 menunjukkan *pixel-pixel* yang ada dalam sebuah gambar.



Gambar 1 : Ilustrasi Pixel pada Citra Digital

### 2. Warna pada Citra Digital

Citra digital memiliki beberapa jenis cara pewarnaan. Tiap jenis pewarnaan ini memiliki karakteristik masing-masing. Jenis pewarnaan ini memberikan pengaruh pada citra digital sehingga memiliki jumlah warna yang berbeda (perbedaan kualitas warna) dan pengaruh pada ukuran dokumen. Berikut adalah jenis-jenis pewarnaan pada citra digital.

#### 1. Hitam dan Putih (*Duotone*)

Pada citra digital dengan pewarnaan *duotone*, warna pada *pixel* hanya memiliki 2 kemungkinan warna, pada umumnya hitam-putih. *Duotone* ini disebut juga dengan warna *1-bit*, karena setiap *pixel* hanya membutuhkan 1 *bit* untuk menyimpan warna *pixel* tersebut. Selain itu *duotone* juga dikenal dengan sebutan *monochrome*.

Dengan penggunaan warna *1-bit*, maka kualitas gambar pada citra digital tidak begitu bagus. Untuk sebuah gambar yang sederhana penggunaan warna *1-bit* sangat sederhana karena memiliki ukuran *file* yang

jauh lebih kecil dibandingkan dengan penggunaan sistem warna lainnya, namun pada sebuah citra digital yang merupakan foto, maka gambar foto tersebut akan mengalami pengurangan kualitas yang signifikan dengan penggunaan warna *1-bit*.



Gambar 2 : Warna 1-bit (256 colors warna)

Sumber : <http://color-me-club.deviantart.com/art/Zebra-linear-99491504>

#### 2. Hitam Putih dan Abu-Abu (*Grayscale*)

Citra digital dengan pewarnaan *grayscale* tidak memiliki warna yang terlalu banyak. Pada *grayscale*, warna yang tersedia hanyalah warna-warna yang ada diantara hitam dan putih, meliputi warna abu-abu yang beragam. Namun kualitas yang dihasilkan warna *grayscale* ini jauh lebih baik dibandingkan dengan *duotone* yang hanya terdiri dari hitam dan putih. Pada gambar dengan pewarnaan *grayscale*, perubahan warna antara satu *pixel* ke *pixel* lainnya tidak terlihat sangat signifikan sehingga gambar dapat lebih mudah dicerna oleh mata manusia. Jika pada *duotone bit* yang dimiliki tiap *pixel* hanya 1, maka pada *grayscale* setiap *pixel* direpresentasikan dalam 8 *bit* data. Dengan penggunaan warna *8-bit* ini maka ukuran dokumennya pun otomatis menjadi semakin besar dibandingkan dengan warna *1-bit*.



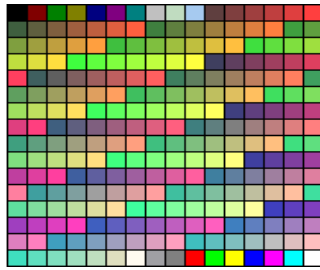
Gambar 3 : Warna Grayscale

Sumber : <http://www.eclipse.org/articles/Article-UI-Guidelines/Index.html>

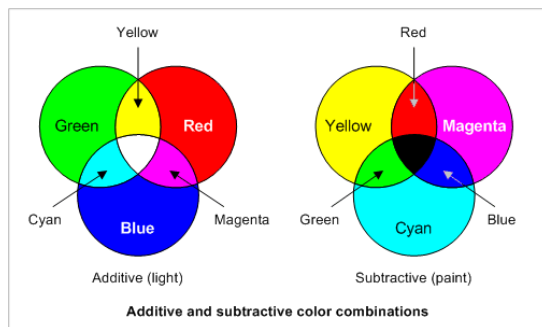
#### 3. Citra Digital Berwarna

Citra digital berwarna membutuhkan ukuran dokumen yang relatif besar jika dibandingkan dengan format *duotone* dan *grayscale* karena kebutuhan penyimpanan *bit* yang lebih banyak. Pada citra digital berwarna, pada umumnya warna *pixel* yang ada terbentuk dari 3 nilai komponen warna, yaitu merah, hijau dan biru (*RGB*).





**Gambar 4 : Warna 8-bit (256 colors warna)**  
Sumber :  
<http://www.eclipse.org/articles/Article-UI-Guidelines/Index.html>



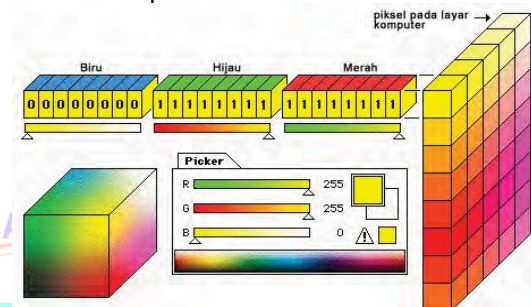
**Gambar 5 : Warna RGB 32-bit**  
Sumber :

<http://www.infotografi.com/blog/2014/06/dynamic-range-bit-depth-dan-color-depth-dalam-sensor-kamera-digital/>

Jumlah *bit* yang digunakan tiap *pixel* pada citra digital berwarna beraneka ragam, mulai dari 8-*bit* hingga 32-*bit*. Semakin besar jumlah *bit* yang digunakan, maka semakin tinggi kualitas warna dari gambar tersebut.

- 8-bit (256 warna)**  
Penggunaan 8-*bit* data untuk tiap *pixel* ini mirip dengan pada pewarnaan *grayscale*, namun 256 warna yang ditentukan tidak semuanya berwarna abu-abu. Kualitas warna dengan pewarnaan 8-*bit* atau yang lebih dikenal dengan sebutan 256 *colors* ini belum terlalu baik. Gambar 2.3 menunjukkan proses menampilkan warna *pixel* pada 256 *colors*.
- Indexed color**  
Pada *indexed color*, jumlah *bit* yang digunakan 8-*bit* dan jumlah warnanya pun 256. Namun 256 warna tersebut dapat didefinisikan dalam citra digital. Dengan mendefinisikan tabel warna yang berisi 256 warna tersebut, kualitas warna gambar dapat meningkat. Gambar 2.4 menunjukkan contoh penggunaan tabel warna untuk meningkatkan kualitas warna pada *indexed color*.
- Red Green Blue (RGB)**

Dengan menggunakan pewarnaan ini, *bit-bit* pembentuk *pixel* merupakan perwakilan dari setiap elemen pembentuk warna yaitu merah, hijau dan biru. Jumlah *bit* yang digunakan pada warna *RGB* sangat beragam, diantaranya adalah 16-*bit*, 24-*bit* dan 32-*bit* (*true color*). Semakin banyak jumlah *bit* yang merepresentasikan warna, maka kualitas warna tersebut semakin baik. Pada warna 32-*bit*, artinya ada 8-*bit* yang merepresentasikan warna merah, 8-*bit* untuk warna biru dan 8-*bit* untuk warna hijau. Gambar 6. menunjukkan bagaimana gambar *RGB* dengan 24-*bit* ditampilkan.



**Gambar 6 : Proses Menampilkan Gambar dengan Warna 24-bit**

### 2.3. Steganografi

Tujuan dari steganografi adalah merahasiakan atau menyembunyikan keberadaan dari suatu pesan tersembunyi atau sebuah informasi. Dalam prakteknya diselesaikan dengan membuat perubahan tipis terhadap digital lain yang isinya tidak akan menarik perhatian dari penyerang potensial, sebagai contoh sebuah gambar yang terlihat tidak berbahaya. Perubahan ini tergantung kunci (sama pada kriptografi) dan pesan untuk disembunyikan. Orang yang menerima gambar kemudian dapat menyimpulkan informasi terselubung dengan cara mengganti kunci yang benar ke dalam algoritma yang digunakan.

Pada steganografi cara ini bermanfaat jika digunakan pada cara steganografi komputer karena banyak *format* digital yang dapat dijadikan media untuk menyembunyikan pesan. *Format* yang umum dilakukan antara lain :

- Format image* : *bitmap* ( *bmp* ), *GIF*, *pcx*, *JPEG*, dll
- Format audio* : *wav*, *voc*, *mp3*, dll
- Format lain* : *txt*, *html*, *pdf*, dll.

### 2.4. Metode Lossy

Lossy kompresi adalah suatu metode untuk mengkompresi data dan mendekompresinya, data yang diperoleh mungkin

berbeda dari yang aslinya tetapi cukup dekat perbedaannya. Lossy kompresi ini paling sering digunakan untuk kompres data multimedia (Audio, gambar diam). Sebaliknya, kompresi lossless diperlukan untuk data teks dan file, seperti catatan bank, artikel teks dll.

Format kompresi lossy mengalami generation lossy yaitu jika melakukan berulang kali kompresi dan dekompresi file akan menyebabkan kehilangan kualitas secara progresif. hal ini berbeda dengan kompresi data lossless. ketika pengguna yang menerima file terkompresi secara lossy (misalnya untuk mengurangi waktu *download*) file yang diambil dapat sedikit berbeda dari yang asli dilevel bit ketika tidak dapat dibedakan oleh mata dan telinga manusia untuk tujuan paling praktis. Metode ini menghasilkan ratio kompresi yang lebih besar daripada metode lossless. Misal terdapat image asli berukuran 12,249 bytes, kemudian dilakukan kompresi dengan JPEG kualitas 30 dan berukuran 1,869 bytes berarti image tersebut 85% lebih kecil dan ratio kompresi 15%. Contoh metode lossy adalah metode CS&Q (*coarser sampling and/or quantization*), JPEG, dan MPEG.

Ada dua skema dasar lossy kompresi :

1. Lossy transform codec, sampel suara atau gambar yang diambil, di potong kesegmen kecil, diubah menjadi ruang basis yang baru, dan kuantisasi. hasil nilai kuantisasi menjadi entropy coded
2. Lossy predictive codec, sebelum dan/atau sesudahnya data di-decode digunakan untuk memprediksi sampel suara dan frame picture saat ini. kesalahan antara data prediksi dan data yang nyata, bersama-sama dengan informasi lain digunakan untuk mereproduksi prediksi, dan kemudian dikuantisasi dan kode.

Dalam beberapa sistem kedua teknik digabungkan, dengan mengubah code yang digunakan untuk mengkompresi kesalahan sinyal yang dihasilkan dari tahapan prediksi, adapun rumus metode lossy dalam mengkompres file citra adalah sebagai berikut :

$$\text{NISBAH} = 100\% - \left( \frac{\text{ukuran\_citra\_hasil\_kompresi}}{\text{ukuran\_citra\_semula}} \right) \times 100\%$$

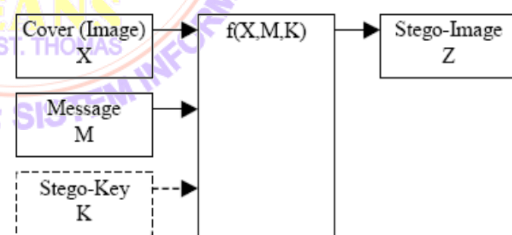
## 2.5. Metode Bit-Plane Complexity Segmentation (BPCS)

*Bit-plane complexity segmentation (BPCS)* adalah salah satu teknik steganografi yang diperkenalkan oleh Eiji Kawaguchi dan R. O. Eason pada tahun 1997. Teknik ini merupakan teknik steganografi kapasitas

besar, karena dapat menampung data rahasia dengan kapasitas yang relatif besar jika dibandingkan dengan metode steganografi lain seperti *LSB (Least Significant Bit)*.

Eiji Kawaguchi dan R. O. Eason memperkenalkan teknik *BPCS* ini untuk digunakan pada dokumen citra dengan format *BMP* yang tidak terkompresi. Dokumen citra tersebut dibagi menjadi beberapa segmen dengan ukuran 8x8 pixel setiap segmennya. Pada dokumen citra 8-bit, setiap satu segmen akan memiliki 8 buah *bitplane* yang merepresentasikan *pixel - pixel* dari setiap *PBC* tersebut. Kedelapan *bitplane* ini disebut *PBC system (Pure Binary Coding)*. Pada *BPCS*, proses *embedding* dilakukan pada *bit-plane* dengan sistem *CGC (Canonical Gray Coding)* karena proses *bit slicing* pada *CGC* cenderung lebih baik dibandingkan pada *PBC*.

Proses penyisipan data dilakukan pada segmen yang memiliki kompleksitas yang tinggi. Segmen yang memiliki kompleksitas tinggi ini disebut *noise-like regions*. Pada segmen-segmen ini penyisipan dilakukan tidak hanya pada *least significant bit*, tapi pada seluruh *bit-plane*. Karena itu kapasitas data pada *BPCS* dapat mencapai 50% dari ukuran *cover-imagenya*. Sumber : Shrikant S. Khaire and DR. Sanjay L. Nalbalwar (2010). "*Review: Steganography – Bit Plane Complexity Segmentation (BPCS) Technique.*" *Department of Electronics & Telecommunication, Dr. Babasaheb Ambedkar Technological University, Lonere, Dist: Raigad, Maharashtra, India. 2(9). 4860-4868.*



**Gambar 7 : Basic digital Steganography Encoder**

Sumber : Shrikant Khaire et. al. / International Journal of Engineering Science and Technology Vol. 2(9), 2010, 4860-4868

### 2.5.1. Kompleksitas Gambar Biner

Kompleksitas gambar biner adalah suatu parameter kerumitan dari suatu gambar biner. Pada makalah ini, ukuran kompleksitas yang akan digunakan adalah *black-and-white border image complexity*. Perubahan warna hitam dan putih dalam gambar biner adalah ukuran yang baik untuk menghitung nilai kompleksitas. Jika perubahan warna yang terjadi banyak, maka gambar tersebut memiliki tingkat kompleksitas

tinggi. Jika sebaliknya, maka gambar tersebut merupakan gambar yang simpel.

Perubahan warna hitam-putih adalah jumlah dari perubahan warna yang terjadi pada setiap baris dan kolom dalam gambar. Sebagai contoh, sebuah *pixel* hitam yang dikelilingi *pixel* putih memiliki nilai perubahan warna 4.

Dengan  $\alpha$  sebagai nilai kompleksitas, maka rumus penghitungan kompleksitas yang akan digunakan adalah :

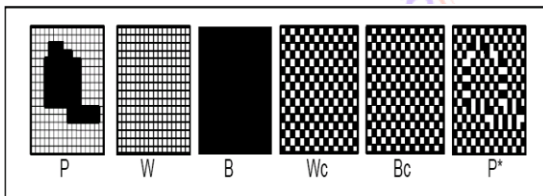
$$\alpha = \frac{k}{n} \quad \dots(1)$$

Dengan  $k$  adalah jumlah perubahan warna hitam-putih dan  $n$  adalah kemungkinan maksimal perubahan warna dalam gambar.

### 2.5.2 Konjugasi dari Gambar Biner

Konjugasi dari suatu gambar biner  $P$  adalah sebuah gambar biner lainnya yang memiliki nilai kompleksitas sebesar satu dikurangi nilai kompleksitas  $P$ . Dari Gambar 1 dapat dilihat bahwa  $P$  adalah gambar yang memiliki *pixel background* dengan pola  $W$  dan *pixel foreground* dengan pola  $B$ .  $P^*$  yang merupakan konjugasi dari  $P$  memiliki spesifikasi sebagai berikut :

1. Memiliki bentuk area *foreground* sama dengan  $P$ .
2. Memiliki pola area *foreground* sama dengan pola  $Bc$ .
3. Memiliki pola area *background* sama dengan pola  $Wc$ .



**Gambar 8 : Proses Menampilkan Gambar dengan Warna 24-bit**

Sumber : Arya Widyanarko, Implementasi Steganografi dengan Metode *Bit-Plane Complexity Segmentation (BPCS)* untuk Dokumen Citra Terkompresi

Untuk membangun sebuah konjugasi  $P^*$  dari sebuah gambar  $P$ , dapat dilakukan dengan rumus berikut, dimana “ $\oplus$ ” menandakan operasi *exclusive OR*.

$$P^* = P \oplus Wc \quad \dots(2)$$

$$(P^*)^* = P \quad \dots(3)$$

$$P^* \neq P \quad \dots(4)$$

Jika  $\alpha(P)$  adalah kompleksitas dari  $P$ , maka:

$$\alpha(P^*) = 1 - \alpha(P) \quad \dots(5)$$

### 2.5.3 Informative Region dan Noise-Like Region

*Informative image* berarti gambar yang simpel, sementara *noise-like region* berarti gambar yang kompleks. Hal ini hanya berlaku pada kasus dimana sebuah gambar biner merupakan bagian dari sebuah gambar yang natural.

Kompleksitas sebuah area *bit-plane* adalah parameter yang digunakan dalam menentukan sebuah *bit-plane* merupakan *informative* atau *noise-like region*. Parameter kompleksitas ini dibatasi oleh nilai *threshold* ( $\alpha_0$ ).

Sebuah *bit-plane* tergolong sebagai *informative region* apabila memiliki nilai kompleksitas yang lebih kecil dibandingkan *threshold* ( $\alpha \leq \alpha_0$ ) dan sebaliknya akan dianggap sebagai *noise-like region*. Pada tugas akhir ini, nilai  $\alpha_0$  yang digunakan beragam, yaitu diantara 0,1 hingga 0,5.

## 3. PEMBAHASAN

### 3.1. Analisis Kecocokan Teknik *Bit-Plane Complexity Segmentation* Citra Terkompresi dengan PNG

*Portable Network Graphics* merupakan *format* yang paling terakhir dikembangkan dibandingkan *GIF* dan *JPEG*. Pengembangannya ditujukan untuk dapat menggantikan peranan *format GIF*. Kedua *format* ini memang memiliki ciri yang cukup mirip, namun *PNG* telah jauh lebih berkembang dengan penggunaan jenis pewarnaan yang lebih beragam.

*Format PNG* secara umum digunakan untuk citra *web* karena dapat menampung *pixel* dengan warna *RGB* 24 dan 48 *bit*, *grayscale* 8 dan 16 *bit* serta *indexed color* dengan kedalaman *pixel* 1 hingga 8 *bit*. Penerapan *BPCS* pada *format* ini tentunya tidak akan memiliki masalah besar dilihat dari aspek pengubahan gambar sebelum dan sesudah penyisipan karena mendukung penggunaan warna *RGB*.

Kemungkinan terbaik pada penerapan *BPCS* ini adalah jika *cover image PNG* menggunakan warna *RGB* atau *grayscale*, bukan *indexed color*. Jika hal tersebut terjadi, maka tidak ada masalah pada penyisipan yang dilakukan. Kemungkinan terburuk adalah jika tabel warna *Digunakan* yang menandakan bahwa *cover image* merupakan citra *PNG* dengan *indexed color* yang memiliki banyak warna yang sangat berbeda-beda. Jika citra tersebut yang *Digunakan* sebagai perantara, maka cukup besar kemungkinan bahwa *stego-image* hasil penyisipan berubah cukup drastis.

Kemungkinan tersebut hanya berkaitan dengan hasil keluaran proses penyisipan yang dikhawatirkan berupa gambar yang rusak. Kemungkinan hilang atau rusaknya pesan



rahasia tidak ada pada penerapan BPCS pada PNG. Hal tersebut cukup jelas mengingat teknik kompresi yang Digunakan pada PNG merupakan kompresi yang *lossless*.

Seperti kedua *format* yang telah dibahas sebelumnya, teknik BPCS pun tidak benar-benar berjalan dengan sempurna pada penerapan penyisipan pada dokumen citra PNG. Kemungkinan rusaknya citra perantara pesan rahasia tetap terbuka walaupun tidak terbuka lebar.

Dengan demikian, maka BPCS dapat diterapkan dengan baik pada citra digital dengan *format PNG* tanpa adanya kerusakan pesan dan hanya kemungkinan kecil menghasilkan *stego-image* yang agak terlihat rusak jika menggunakan *indexed color*. *Format* ini cocok dengan teknik BPCS.

**Tabel 1 : Perbandingan Kecocokan BPCS dengan JPEG dan PNG**

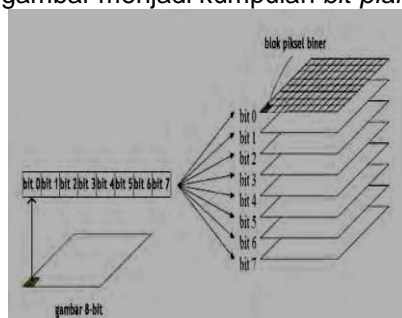
ASPEK	JPEG	GIF	PNG
Ada bit yang hilang pada saat kompresi	Ya	Tidak	Tidak
Pesan rahasia terjaga tanpa ada kerusakan	Tidak	Ya	Ya
Mendukung RGB atau <i>indexed color</i>	RGB	<i>Indexed color</i>	Keduanya
Kecocokan dengan BPCS ( skala 0-5 )	4	2,5	4,5

Tabel 1. menunjukkan perbandingan kecocokan ketiga *format* citra terkompresi yang dianalisis dengan BPCS. *Format PNG* merupakan *format* yang paling cocok dengan BPCS.

### 3.2. Analisis Algoritma BPCS

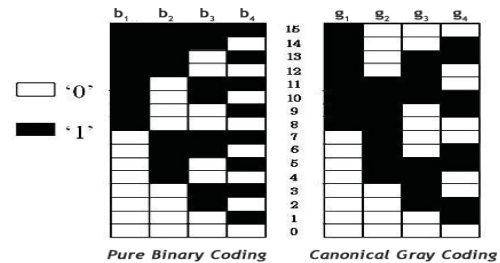
Berikut adalah langkah-langkah yang dilakukan pada algoritma BPCS saat menyisipkan data :

1. Mengubah *cover image* dari sistem PBC menjadi sistem CGC. Sebelumnya, gambar tersebut di-slice terlebih dahulu menjadi *bit-plane*. Setiap *bit-plane* mewakili *bit* dari setiap *pixel*. Gambar 9 menunjukkan proses pemotongan gambar menjadi kumpulan *bit-plane*.



**Gambar 9 : Bit-Slicing pada Gambar 8-bit**

Jika dilihat dari gambar 9, maka *bit-plane* yang disusun oleh *bit 7* merupakan *least significant bit*. Pada BPCS, semua *bit-plane* memiliki kemungkinan untuk diganti, tidak hanya *least significant bit*. *Bit-plane* yang berupa gambar *biner* tersebut dikonversi dari PBC menjadi CGC. Gambar 12 menunjukkan perbedaan pada PBC dan CGC.



**Gambar 10 : Gambar Biner Dengan PBC dan CGC**

2. Segmentasi setiap *bit-plane* pada *cover image* menjadi *informative* dan *noise like region* dengan menggunakan nilai batas/*threshold* ( $\_0$ ). Nilai umum dari  $\_0=0,4$ .
3. Bagi setiap *byte* pada data rahasia menjadi blok-blok
4. Jika blok(S) tidak lebih kompleks dibandingkan dengan nilai batas, maka lakukan konjugasi terhadap S untuk mendapatkan  $S^*$  yang lebih kompleks.
5. Sisipkan setiap blok data rahasia ke *bit-plane* yang merupakan *noise-like region* (atau gantikan semua *bit* pada *noise-like region*). Jika blok S dikonjugasi, maka simpan data pada "*conjugation map*".
6. Sisipkan juga pemetaan konjugasi yang telah dibuat
7. Ubah *stego-image* dari sistem CGC menjadi sistem PBC
8. Proses ekstraksi data rahasia dapat dilakukan dengan menerapkan langkah-langkah penyisipan secara terbalik. Sebagai contoh, sebuah dokumen citra akan disisipi sebuah pesan rahasia Ms. Pertama-tama *pixel* pada citra tersebut (*cover image*) dibagi menjadi segmen-segmen gambar *biner* seperti ditunjukkan pada gambar 2.9. Kemudian pesan rahasia dibagi menjadi blok yang masing-masing berukuran 64 *bit*, dan direpresentasikan pada matriks berukuran 8x8.

Dari langkah tersebut, hasilnya kita mendapatkan *stego-object*. Lalu cara mengekstrak informasinya adalah dengan:

1. Ubah *stego-object* dari representasi PBC ke CGC

2. Tentukan segmen mana yang informatif dan segmen mana yang noise-like dengan *threshold* yang sama dengan *threshold* ketika proses *embedding*. Dengan kata lain nilai Ukuran berkas original adalah 944,408 *byte*. Ternyata *BPCS* mampu menampung berkas berukuran 441,418 *byte* dalam *cover-object*. Dengan kata lain kapasitas penyimpanannya mencapai 40% lebih. Selain itu jika kita hitung lagi bahwa data yang ditanam adalah hasil kompresi dari ukuran 1,212,744 *byte*, hasil steganografi ini terasa menakjubkan. Lebih baik lagi jika hasil *stego-object* dikompresi dengan kompresi yang *lossless*. Pada simulasi ini, kompresi dilakukan sampai *stego-object* berukuran 505,502 *byte*. Dengan kata lain data yang disimpan dalam *stego-object* berukuran 505,502 *byte* ini adalah sebanyak data *embedded message* yang belum dikompresi yaitu sebesar 1,212,744 *byte*. Pesan yang disimpan mencapai 2.5x ukuran *stego-object*.

Algoritma untuk *decoding* hanyalah prosedur kebalikan dari langkah memasukkan kode. Inovasi *steganography BPCS* adalah sebagai berikut:

1. Pembagian tiap *bit-plane* dari gambar berwarna ke dalam daerah "informatif" dan "noise like"
2. Pengenalan batas BW ( *black - white*) berdasarkan pengukuran kompleksitas ( $\alpha$ ) untuk pembagian daerah.
3. Pengenalan operasi konjugasi untuk mengubah blok rahasia sederhana menjadi blok yang kompleks.
4. Menggunakan *image plane CGC* dan bukan *image plane PBC*.

Metode *steganography BPCS* memiliki beberapa parameter untuk penerapan program praktis. Berapa di antaranya adalah :

1. Lokasi *header* dari *file* rahasia
2. Parameter enkripsi dari *file* rahasia
3. Parameter kompresi dari *file* rahasia.

### 3.3 Analisa Metode Lossy

Tabel 3 : Matrik RGB Sebelum dikompres

1 4 7 001 100 111	1 4 7 001 100 111	3 6 7 011 110 111	7 1 5 111 001 101	1 2 5 001 010 101	2 5 1 010 101 001
4 7 6 100 111 110	4 7 4 100 111 100	6 7 4 110 111 100	1 5 4 001 101 100	2 5 4 010 101 100	2 5 2 010 101 010
7 6 5 111 110 101	7 4 5 111 100 101	7 4 2 111 100 010	5 4 2 101 100 010	5 2 2 101 010 010	1 2 1 001 010 001
6 5 2 110 101 010	4 5 2 100 101 010	4 2 3 100 010 011	4 2 3 100 010011	2 2 0 010 010 000	2 1 0 010 001 010
5 2 1 101 010 001	5 2 1 101 010 001	2 3 3 010 011 001	2 3 3 010 011 011	2 0 1 010 000 001	1 1 2 001 001 010
2 1 4 010 001 100	2 1 4 010 001 100	3 3 6 011 011 110	3 7 1 001 111 001	0 1 2 000 001 010	0 2 2 000 010 010

Untuk mengetahui bagaimana proses metode *lossy* dalam mengkompres file citra atau gambar, berikut adalah analisisnya disertakan dengan sampel gambar yang akan di proses.



Gambar 11 : Citra Sebelum Di Kompresi

#### Contoh:

Misalnya sebuah citra dengan memiliki image size ukuran 6x6 *pixel* dengan 8 derajat R, 8 derajat G dan 8 derajat B sebagai berikut:

Matrik R:	Matrik G:	Matrik B:
1 1 3 7 1 2	4 4 6 1 2 2	7 7 7 5 5 1
4 4 6 1 2 2	7 7 7 5 5 1	6 4 4 4 2 2
7 7 7 5 5 1	6 4 4 4 2 2	5 5 2 2 2 1
6 4 4 4 2 2	5 5 2 2 2 1	2 2 3 3 0 0
5 5 2 2 2 1	2 2 3 3 0 0	1 1 3 7 1 2
2 2 3 3 0 0	1 1 3 7 1 2	4 4 6 1 2 2

Tabel 2 : Penggabungan Matriks RGB sebelum dikompres

RGB	RGB	RGB	RGB	RGB	RGB
1 4 7	1 4 7	3 6 7	7 1 5	1 2 5	2 5 1
4 7 6	4 7 4	6 7 4	1 5 4	2 5 4	2 5 2
7 6 5	7 4 5	7 4 2	5 4 2	5 2 2	1 2 1
6 5 2	4 5 2	4 2 3	4 2 3	2 2 0	2 1 0
5 2 1	5 2 1	2 3 3	2 3 3	2 0 1	1 1 2
2 1 4	2 1 4	3 3 6	3 7 1	0 1 2	0 2 2

Dan dapat diturunkan kedalam bilangan Biner sebagai berikut:



Yang akan dikompres menjadi citra dengan 4 derajat keabuan (0 s/d 3), jadi setiap derajat keabuan direpresentasikan dengan 2 bit. Adapun langkah-langkah untuk mengkompres dengan menggunakan metode lossy yaitu:

1. Langkah I: Buat Histogram citra semula Histogram citra semula untuk menghasilkan nilai matriks

0	*****
1	*****
2	*****
3	*****
4	*****
5	*****
6	*****
7	*****

Ada 108 *pixel*, dikelompokkan menjadi 4 kelompok derajat RGB. tiap kelompok ada sebanyak rata-rata  $108/4 = 27$  *pixel* perkelompok.

23	0	*****
	1	*****
39	2	*****
	3	*****
28	4	*****
	5	*****
17	6	*****
	7	*****

7 \*\*\*\*\*

2. Langkah yang ke II menggunakan tabel RGB 8 tingkat dan RGB 4 tingkat.

Di bawah ini akan digunakan tabel untuk menghitung nilai matriks hasil kompres dan sebelum dikompres dengan menggunakan tabel derajat RGB Lama 8 tingkat dan derajat RGB baru 4 tingkat.

**Tabel 4 : Perubahan Derajat RGB Lama Menjadi RGB Baru**

Derajat RGB lama 8 Tingkat	Derajat RGB baru 4 Tingkat
0	0
1	0
2	1
3	1
4	2
5	2
6	3
7	3

Matriks citra kompresi adalah sebagai berikut :

Matrik R :	Matrik G:	Matrik B:
001301	223011	333220
223011	333220	322211
333220	322211	221110
322211	221110	111100
221110	111100	001301
111100	001301	223011

**Tabel 5 : Penggabungan RGB Setelah Dikompres**

R G B	RGB	RGB	RGB	RGB	RGB
0 2 3	0 2 3	1 3 3	3 0 2	0 1 2	1 1 0
2 3 2	2 3 2	3 3 2	0 2 2	1 2 1	1 0 1
3 3 2	3 2 2	3 2 1	2 2 1	2 2 1	0 1 0
3 2 1	2 2 1	2 1 1	1 1 3	1 1 0	1 0 0
2 1 0	2 1 0	1 1 0	1 1 1	1 0 0	0 0 1
1 0 2	1 0 2	1 1 3	1 3 0	0 0 1	0 1 1

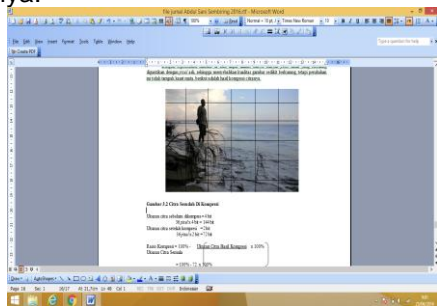
Dan penggabungan RGB dapat diturunkan kedalam Biner sebagai berikut:

**Tabel Bilangan Biner 6 : Matrik RGB Sebelum dikompres**

0 2 3 000 010 011	0 2 3 000 010 011	1 3 3 001 001 011	3 0 2 011 000 010	0 1 2 000 001 012	1 1 0 001 001 000
2 3 2 010 011 010	2 3 2 010 001 010	3 3 2 011 011 001 010	0 2 2 000 010 010	1 2 1 001 012 001	1 0 1 001 000 001
3 3 2 011 001 010	3 2 2 011 001 010	3 2 1 011 001 001	2 2 1 001 010 001	2 2 1 001 010 001	0 1 0 000 001 000

3 2 1 011 001 001	2 2 1 001 010 001	2 1 1 010 001 001	1 1 3 001 011 011	1 1 0 001 001 000	1 0 0 001 000 000
2 1 0 001 001 000	2 1 0 001 001 000	1 1 0 001 001 000	1 1 1 001 001 001	1 0 0 001 000 000	0 0 1 000 000 001
1 0 2 001 000 001	1 0 2 001 000 001	1 1 3 001 001 011	1 3 0 001 011 000	0 0 1 000 000 001	0 1 1 000 001 001

Dengan representasi matriks di atas dapat dilihat bahwa banyak *pixel* lama yang terbuang digantikan dengan *pixel* asli, sehingga menyebabkan kualitas gambar sedikit berkurang, tetapi perubahan ini tidak tampak kasat mata, berikut adalah hasil kompresi citranya.



**Gambar 13 : Citra Sesudah Di Kompresi**

Ukuran citra sebelum dikompres = 4 bit  
 $36 \text{ pixel} \times 4 \text{ bit} = 144 \text{ bit}$   
 Ukuran citra setelah kompresi = 2 bit  
 $36 \text{ pixel} \times 2 \text{ bit} = 72 \text{ bit}$

Rasio Kompresi =  $100\% - \left( \frac{\text{Ukuran Citra Hasil}}{\text{Ukuran Citra Semula}} \right) \times 100\%$   
 $= 100\% - \frac{72}{144} \times 100\%$   
 $= 50\%$ , yang artinya 50% dari citra semula telah dikompres.

**KESIMPULAN**

1. Kapasitas hasil steganografi ( *format* PNG ) akan lebih besar beberapa kali dari gambar awal yang ber-*format* JPG
2. Kapasitas hasil steganografi ( *format* PNG ) akan sama dari gambar awal yang ber-*format* PNG.
3. Steganografi dengan metode *BPCS* dapat diterapkan pada citra terkompresitanpa adanya modifikasi pada proses utama penyisipan dan ekstraksi pesan
4. Tingkat hasil kompresi dengan menggunakan Metode Lossy tergantung pada format file gambar yang dikompres.
5. Perangkat lunak ini dapat menampilkan hasil kompresi berupa ukuran awal dan ukuran sesudah dikompresi.

**DAFTAR PUSTAKA**

- [1]. Abdul Sani Sembiring, “Analisis Steganografi Pada Citra Terkompresi Dengan Metode Bit-Plane Complexity Segmentation (BPCS), Cemerlang 0216 – 3535, Vol 12/1-117/NOP-DES 2011
- [2]. Eiji Kawaguchi and Richard O. Eason. “Principle and applications of BPCS-Steganography”, Kyushu Institute of Technology, Kitakyushu, Japan and University of Maine, Orono, Maine 04469-5708.
- [3]. Shrikant S. Khaire and DR. Sanjay L. Nalbalwar (2010). "Review: Steganography – Bit Plane Complexity Segmentation (BPCS) Technique", Department of Electronics & Telecommunication, Dr. Babasaheb Ambedkar Technological University, Lonere, Dist: Raigad, Maharashtra, India. 2(9). 4860-4868.
- [4]. Sherin Jabbar and Shaiju Panchikkil. “Lossy Compression Schemes Based On Transforms-A Literature Review On Medical Images”, Department of Computer Engineering, MES Engineering College, Kuttippuram, Kerala, International Journal of Advanced Information Technology (IJAIT) Vol. 2, No.6, December 2012
- [5]. Mulyanto Edy, S.Si., M.Kom. (2009), “ Teori Pengolahan Citra”, Jogjakarta : Penerbit Andi.
- [6]. Miano and John (1999). “Compressed Image File Format JPEG, PNG, GIF, XBM, BMP” Addison Wesley Longman, Inc.
- [7]. Negnevitsky Michael (2002), “*Artificial Intelligence : A Guide to Intelligence System*”, Enggland: Addison-Wesley.
- [8]. Dony Ariyus (2006),”*Computer Security*”, Yogyakarta :Andi Offset.
- [9]. <http://color-me-club.deviantart.com/art/Zebra-lineart-99491504>
- [10]. <http://www.eclipse.org/articles/Article-UI-Guidelines/Index.html>
- [11]. <http://www.infotografi.com/blog/2014/06/dynamic-range-bit-depth-dan-color-depth-dalam-sensor-kamera-digital/>