

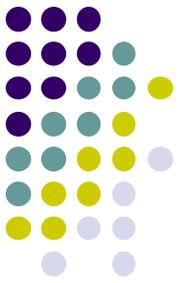
Metode BPCS

(Bit-Plane Complexity Segmentation)

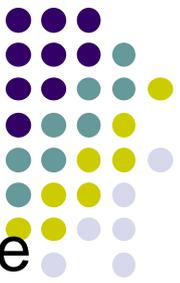
Oleh: Dr. Rinaldi Munir

Program Studi Informatika
Sekolah Teknik Elektro dan Informatika ITB

BPCS



- BPCS = *Bit-Plane Complexity Segmentation*.
- Dikembangkan oleh Eij Kawaguchi dan R. O. Eason pada tahun 1997.
- Merupakan metode steganografi berkapasitas besar, lebih besar daripada menggunakan metode modifikasi LSB.
- Jika metode modifikasi LSB hanya dapat menyisipkan pada satu atau beberapa bit LSB, maka metode BPCS menyisipkan pada satu *bit-plane*.



- Ingat kembali teori yang telah dijelaskan pada metode modifikasi LSB.
- *Bit-plane* adalah citra biner yang berisi bit ke- i dari pixel-pixel di dalam citra.
- Pada citra grayscale 1 *pixel* = 1 *byte*, dan setiap *byte* panjangnya 8 bit (bit ke-1 sampai bit ke-8), maka terdapat 8 buah *bit-plane*.
- Pada citra berwarna 24-bit, 1 *pixel* = 3 *byte*, maka terdapat 24 buah *bit-plane*.



- Contoh 8 buah *bit-plane* pada citra *grayscale*.



Original image



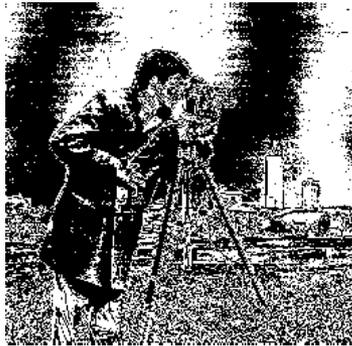
Bit-plane 7



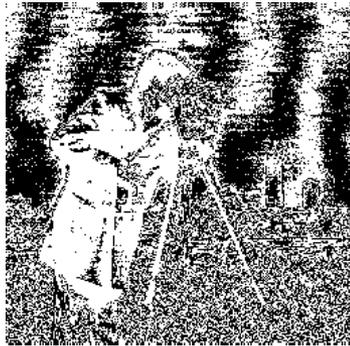
Bit-plane 6



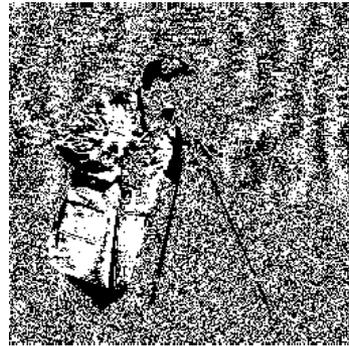
Bit-plane 5



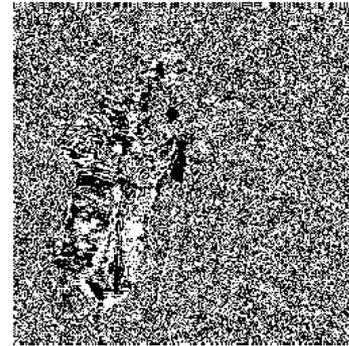
Bit-plane 4



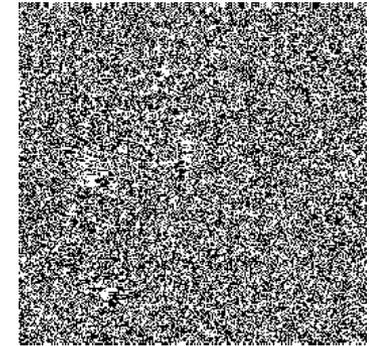
Bit-plane 3



Bit-plane 2



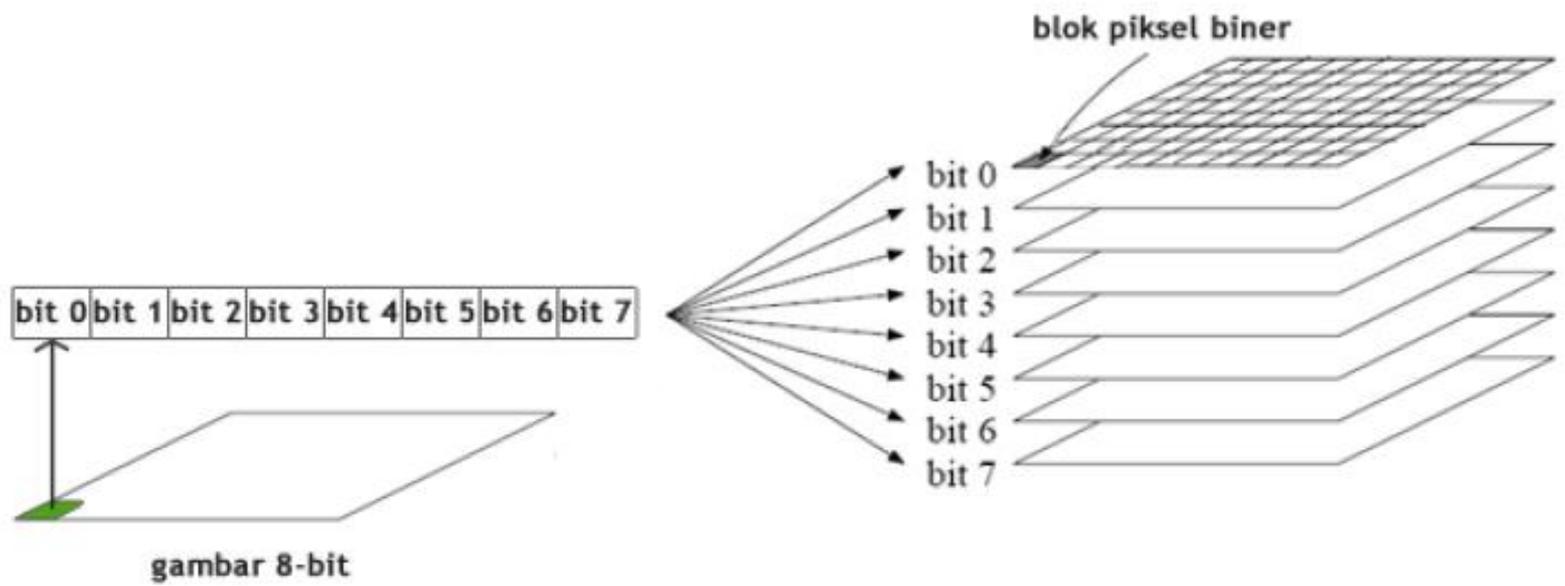
Bit-plane 1



Bit-plane 0



- Pada metode BPCS, citra dibagi menjadi blok berukuran 8×8 *pixel*. Setiap blok *pixel* memiliki 8 buah *bit-plane*. Setiap *plane* berisi $8 \times 8 = 64$ bit.
- Delapan buah *bit-plane* tersebut dinamakan *PBC system* (*Pure Binary Coding*).



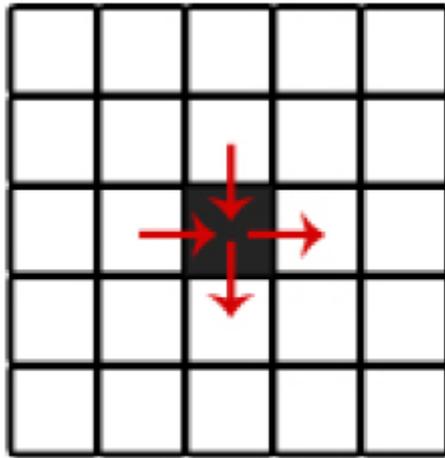


- Penyisipan bit-bit pesan dilakukan pada *bit-plane* yang memiliki kompleksitas tinggi, yang disebut *noise-like regions*.
- Penyisipan pesan dilakukan pada seluruh bit di *noise-like regions* (64 bit), sehingga bit yang disisipkan jauh lebih banyak daripada menggunakan metode modifikasi LSB.
- Karena itu, kapasitas data pada BPCS dapat mencapai 50% dari ukuran *cover-image*.

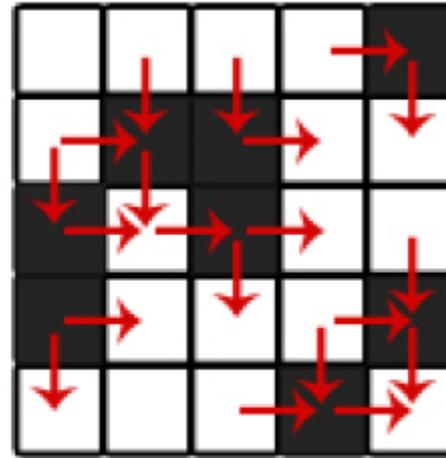


Kompleksitas Citra Biner

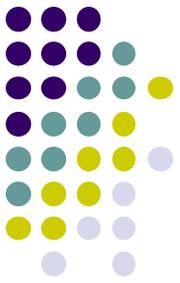
- Kompleksitas citra biner adalah ukuran kerumitan dari suatu citra biner.
- Eiji Kawaguchi mendefinisikan ukuran kompleksitas sebagai *black-and-white border image complexity*.
- *Black-and-white border image complexity* dihitung dari jumlah perubahan warna hitam dan putih. Jika jumlah perubahan warna yang terjadi banyak maka dikatakan citra tersebut memiliki kompleksitas tinggi.



(a)



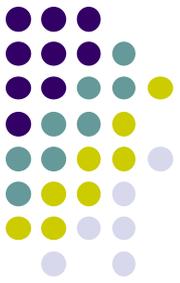
(b)



- Pada gambar (a), jumlah perubahan warna hitam dan putih = 4.
- Pada gambar (b), jumlah perubahan warna hitam dan putih = 20.
- Kompleksitas (α) dihitung dengan rumus $\alpha = \frac{k}{n}$

yang dalam hal ini, k = jumlah perubahan warna hitam dan putih dan n = jumlah kemungkinan perubahan warna di dalam citra.

Canonical Gray Coding (CGC)



- Proses penyisipan pesan pada *bit-plane* lebih baik dilakukan dengan sistem CGC (*Canonical Gray Coding*) ketimbang dengan sistem *PBC*.

- Mengubah sistem *PBC* menjadi sistem CGC dilakukan dengan persamaan *XOR* berikut:

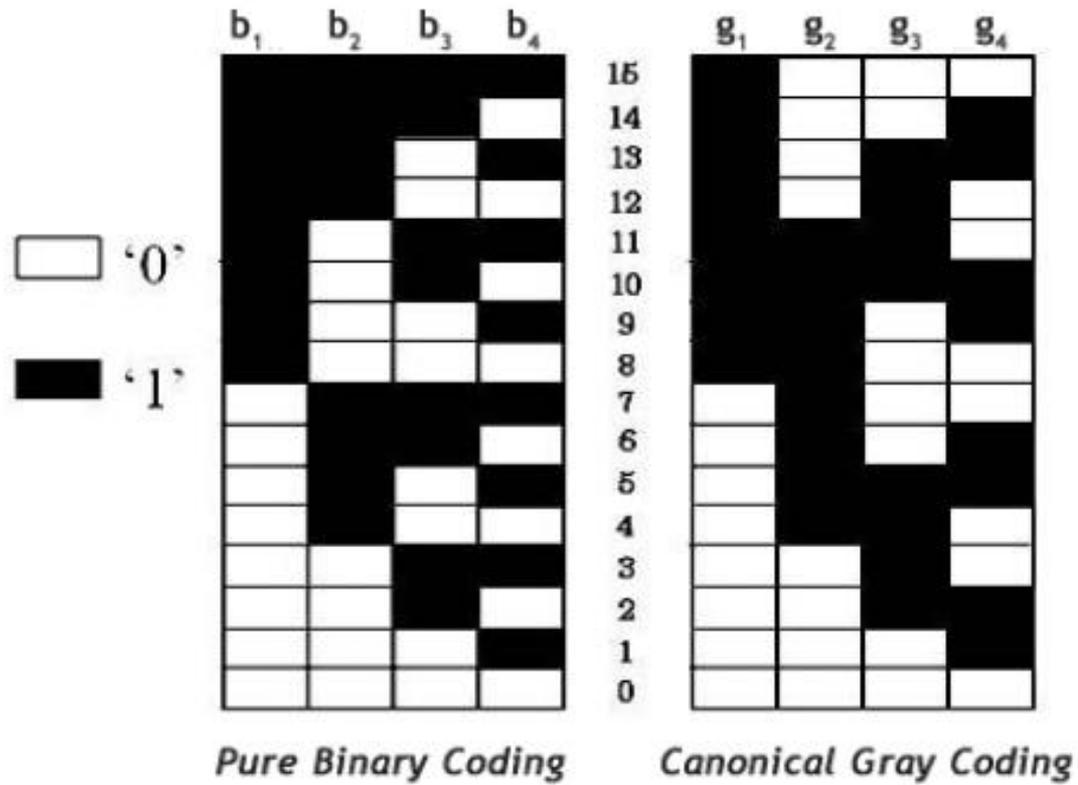
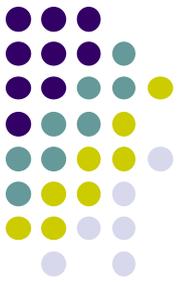
$$g_1 = b_1$$

$$g_i = b_{i-1} \oplus b_i$$

- Sebaliknya, mengubah sistem CGC menjadi *PBC* dilakukan dengan persamaan berikut:

$$b_1 = g_1$$

$$b_i = b_{i-1} \oplus g_i$$



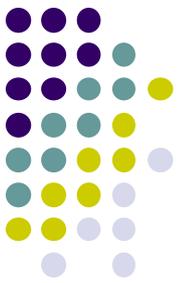
Informative Region dan Noise-like Region



- *Informative region*: bagian citra yang berisi gambar yang simpel.
 - bagian penting, sensitif terhadap manipulasi
 - tidak dilakukan penyisipan pesan di sini
- *Noise-like region*: bagian citra yang berisi gambar yang kompleks.
 - bagian yang kurang informatif.
 - perubahan bit akibat manipulasi tetap membuatnya bersifat *noise-like region*
 - penyisipan pesan dilakukan di sini.

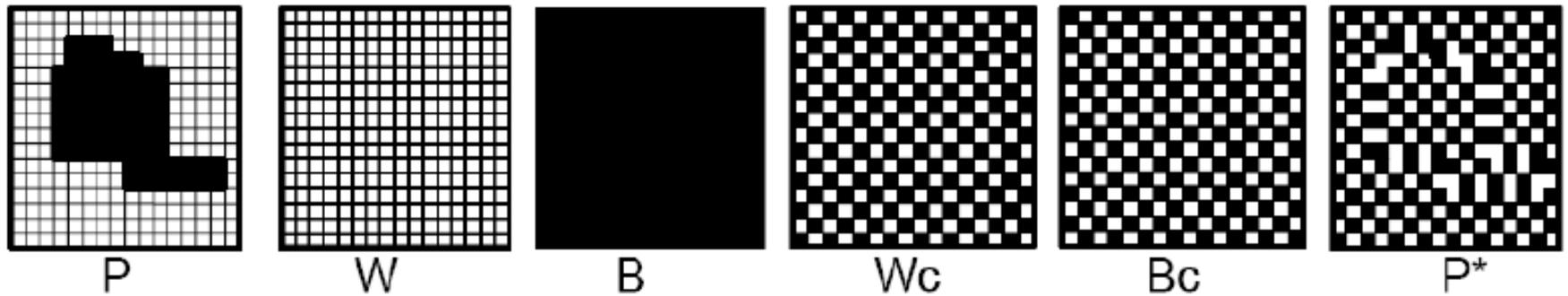


- Untuk menentukan apakah sebuah *region* termasuk kategori *informative* atau kategori *noise-like*, digunakan parameter kompleksitas.
- Sebuah *region* dimasukkan sebagai kategori *informative* jika nilai kompleksitasnya **lebih kecil** dari suatu nilai ambang (*threshold*), α_0 .
- Sebuah *region* dimasukkan sebagai kategori *noise-like* jika nilai kompleksitasnya **lebih besar** dari suatu nilai ambang (*threshold*), α_0 .
- Nilai ambang yang digunakan bervariasi antara 0.1 sampai 0.5



Konyugasi Citra Biner

- Sebuah citra biner 8×8 *pixel* dapat dikonyugasi dengan citra biner lain berukuran sama.
- Tujuan: meningkatkan nilai kompleksitas.
- Jika citra biner P memiliki kompleksitas α , maka konyugasinya memiliki kompleksitas $1 - \alpha$.
- Contoh: misalkan terdapat 5 citra biner sebagai berikut:
 - P : citra biner dengan *background* putih dan *foreground* hitam
 - W : citra biner dengan semua pixel berwarna putih
 - B : citra biner dengan semua pixel berwarna hitam
 - Wc : citra biner dengan pola papan catur
 - Bc : citra biner dengan pola papan catur, negasi dari Wc .



- P^* adalah konyugas dari P dengan 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

- Jadi, untuk membangun konyugasi P^* dari P , caranya:

$$P^* = P \oplus Wc$$

- Perhatikan bahwa

$$(P^*)^* = P$$

$$P^* \neq P$$

$$\alpha(P^*) = 1 - \alpha(P)$$

Algoritma BPCS

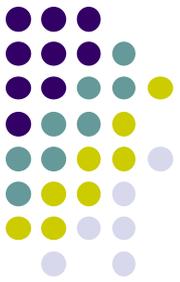


Asumsikan citra berukuran kelipatan 8. Jika bukan kelipatan 8, tambahkan *pixel-pixel* semu sehingga berukuran 8 x 8.

1. Bagi *cover-image* menjadi blok 8 x 8 *pixel*.
2. Bentuk setiap blok 8 x 8 *pixel* menjadi sistem PBC yang terdiri dari 8 buah *bit-plane*.
3. Ubah sistem *PBC* menjadi sistem *CGC* (*Canonical Gray Coding*) → Opsional.
4. Tentukan apakah setiap *bit-plane* merupakan *informative region* atau *noise-like region* dengan menggunakan nilai ambang α_0 . Nilai *default* $\alpha_0 = 0.3$. Jika tergolong *noise-like region*, maka pesan bisa disisipkan pada *bit-plane* tersebut, tetapi jika termasuk *informative region*, maka tidak dapat digunakan untuk menyisipkan pesan.



5. Bagi pesan menjadi segmen-segmen berukuran 64-bit, lalu nyatakan segmen menjadi blok biner berukuran 8×8 .
6. Jika blok pesan S tidak lebih kompleks dibandingkan dengan nilai ambang α_0 (yaitu termasuk kategori *informative region*), lakukan konyugasi terhadap S untuk mendapatkan S^* yang lebih kompleks.
7. Sisipkan segmen pesan 64-bit ke *bit-plane* yang merupakan *noise-like region* dengan cara mengganti seluruh bit pada *noise-like region tersebut* dengan 64-bit pesan).
8. Jika bloks S dikonyugasi, simpan pesan pada “conjugation map”.
9. Sisipkan juga pemetaan konyugasi yang telah dibuat.
10. Ubah *stego-image* dari sistem *CGC* menjadi sistem *PBC*.



Ekstraksi Pesan

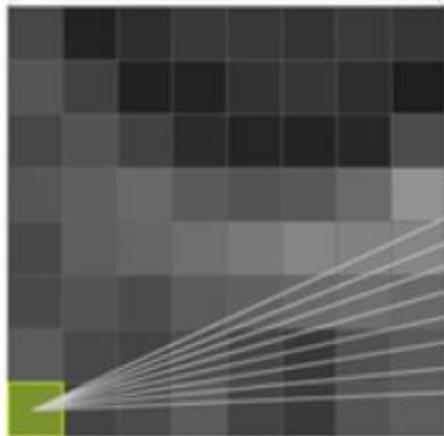
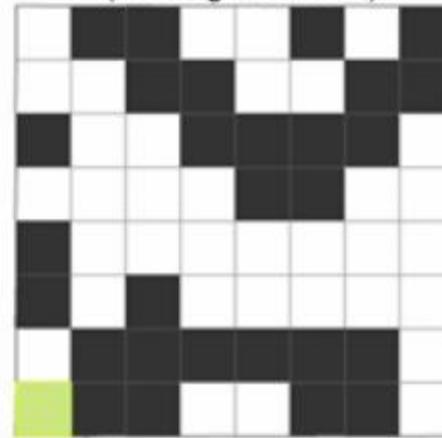
1. Bagi *stego-image* menjadi blok 8 x 8 *pixel*.
2. Bentuk setiap blok 8 x 8 *pixel* menjadi sistem *PBC* yang terdiri dari 8 buah *bit-plane*.
3. Ubah sistem *PBC* menjadi sistem *CGC* (*Canonical Gray Coding*) → Opsional.
4. Hitung kompleksitas setiap *bit-plane*. Jika kompleksitasnya di atas nilai ambang α_0 , maka *bit-plane* tersebut bagian dari pesan. Tabel konyugasi yang disisipkan juga dibaca untuk melihat proses konyugas yang perlu dilakukan pada tiap blok pesan.

Contoh:

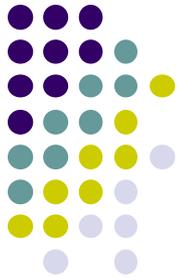
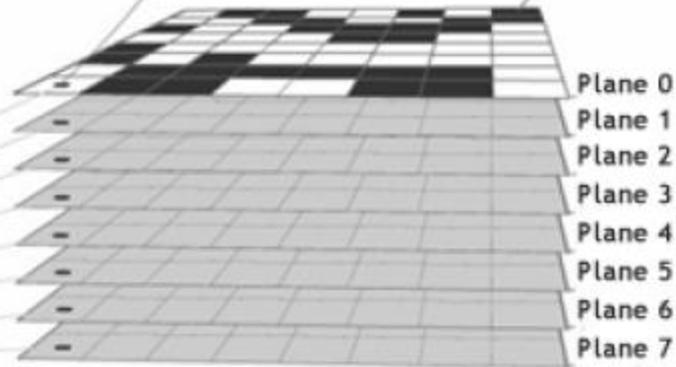
Gambar 8-bit dalam potongan 8x8 piksel



Gambar biner dari plane 0
(Most Significant bit)



Potongan 8x8 piksel





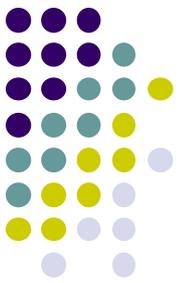
Pada *bit-plane* 0, jumlah perubahan warna hitam-putih adalah 47 kali. Jadi $k = 47$.

Jumlah maksimum perubahan warna pada citra biner yang berukuran 8×8 adalah 112 kali. Jadi $n = 112$.

Nilai kompleksitas *bit-plane* 0 adalah $\alpha = 47/112 = 0.42$.

Dengan menggunakan nilai ambang $\alpha_0 = 0.3$, maka $0.42 > \alpha_0$, sehingga *bit-plane* 0 dikategorikan sebagai *noise-like region* sehingga bisa digunakan untuk menyisipkan pesan.

Bit-plane 1 juga termasuk *noise-like region* karena nilai kompleksitasnya lebih besar dari α_0 .



- Misalkan pesan M yang akan disisipkan adalah sepanjang 128 bit sebagai berikut:

$M_S =$ 1100000101010001010101000000001111010000000111011101111001000100
1100000100000000111110001111100011000011110000110000000000111111

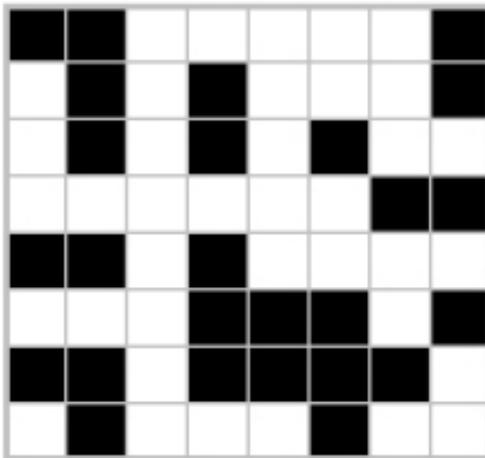
- Bagi pesan M menjadi dua buah blok, S_0 dan S_1 .

$M_{S0} =$ 1100000101010001010101000000001111010000000111011101111001000100
 $M_{S1} =$ 1100000100000000111110001111100011000011110000110000000000111111



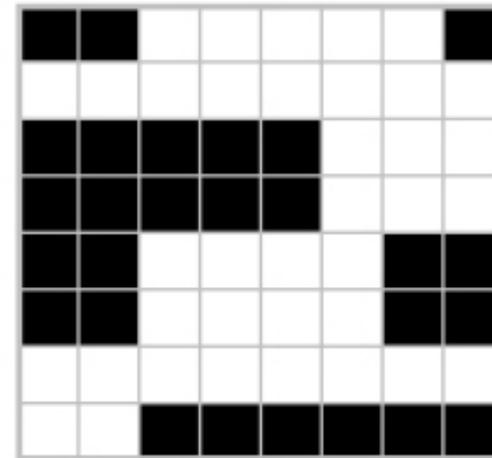
- Representasikan setiap blok pesan menjadi citra biner yang berukuran 8 x 8.

$M_{50} =$ 11000001 01010001
01010100 00000011
11010000 00011101
11011110 01000100



Blok pertama pesan

$M_{51} =$ 11000001 00000000
11111000 11111000
11000011 11000011
00000000 00111111



Blok kedua pesan



Penyisipan blok pesan M_{S_0}

Sebelum disisipkan, hitung terlebih dahulu kompleksitas blok pesan M_{S_0} .

Jumlah perubahan warna = 54 kali

Kompleksitas $\alpha = 54/112 = 0.48$.

Karena $0.48 > \alpha_0$, maka semua *bit plane* 0 diganti dengan 64-bit blok pesan M_{S_0} .

Penyisipan blok pesan M_{S_1}

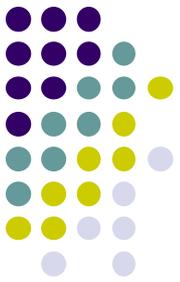
Sebelum disisipkan, hitung terlebih dahulu kompleksitas blok pesan M_{S_1} .

Jumlah perubahan warna = 32 kali

Kompleksitas $\alpha = 32/112 = 0.29$.

Karena $0.29 < \alpha_0$, itu berarti blok pesan M_{S_1} tidak cukup kompleks. Agar cukup kompleks, lakukan konyugasi terhadap M_{S_1} . Misalkan hasil konyugasi adalah $M_{S_1^*}$ dan nilai kompleksitasnya 0.71. Karena $0.71 > \alpha_0$, maka semua *bit plane* 1 diganti dengan 64-bit blok pesan $M_{S_1^*}$.

Referensi



1. Arya Widyanarko, *Implementasi Stegtanografi dengan Metode Bit-Plane Complexity Segmentation (BPCS) untuk Dokumen Citra Terkompresi*, Tugas Akhir Informatika ITB, 2008.