

# Pengenalan *Watermarking*

Bahan Kuliah IF4020 Kriptografi

Oleh: Rinaldi Munir



Program Studi Informatika  
Sekolah Teknik Elektro dan Informatika  
ITB

# Fakta



- Jutaan gambar/citra digital bertebaran di internet via *email, website, bluetooth*, dsb
- Siapapun bisa mengunduh citra, meng-*copy*-nya, menyunting, mengirim, memanipulasi, dsb.
- Sekali citra diunduh/*copy*, referensi pemilik yang asli hilang → sebab informasi kepemilikan tidak melekat di dalam citra.
- Memungkinkan terjadi pelanggaran HAKI:
  - mengklaim kepemilikan citra orang lain
  - pemilik citra yang asli tidak mendapat royalti atas penggandaan ilegal

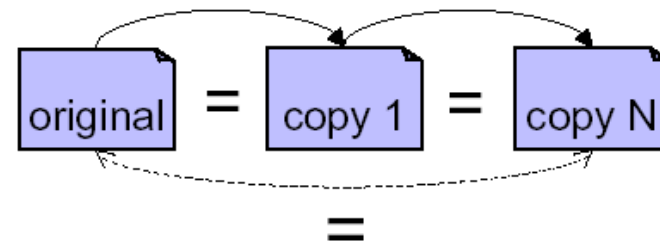
# Siapa pemilik citra ini?





# Karakteristik Dokumen Digital

- Dokumen digital
  - citra (*JPEG/GIF/BMP/TIFF Images*)
  - audio (*MP3/WAV audio*)
  - video (*MPEG video*)
  - teks (*Ms Word document*)



Kelebihan dokumen digital (sekaligus kelemahannya):

- Tepat sama kalau digandakan
- Mudah didistribusikan (misal: via internet)
- Mudah di-edit (diubah)
- Tidak ada perlindungan terhadap kepemilikan, *copyright*, *editing*, dll.
- Solusi: ***digital watermarking***.

# Digital Watermarking



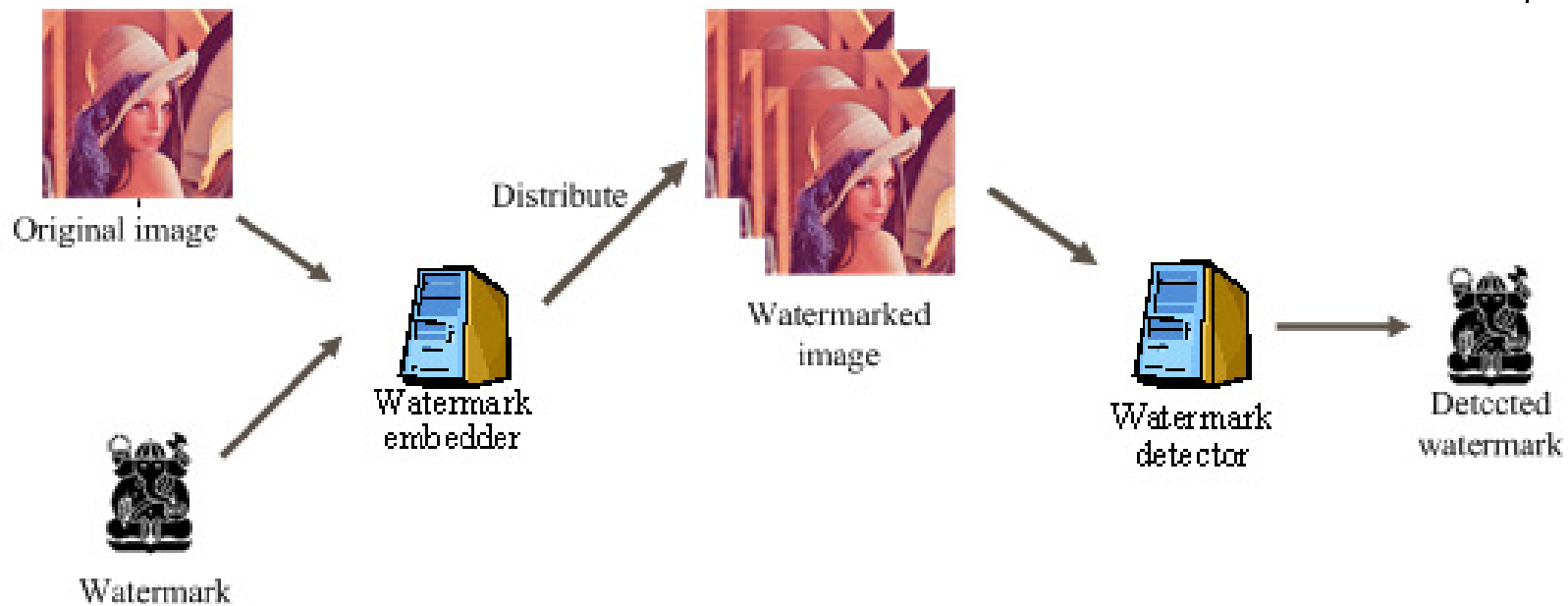
- *Digital Watermarking*: penyisipan informasi (*watermark*) yang menyatakan kepemilikan data multimedia
- *Watermark*: teks, logo, audio, data biner (+1/-1), barisan bilangan riil
- *Watermarking* merupakan aplikasi steganografi
- Tujuan: memberikan perlindungan *copyright*



+ shanty =



# Image Watermarking

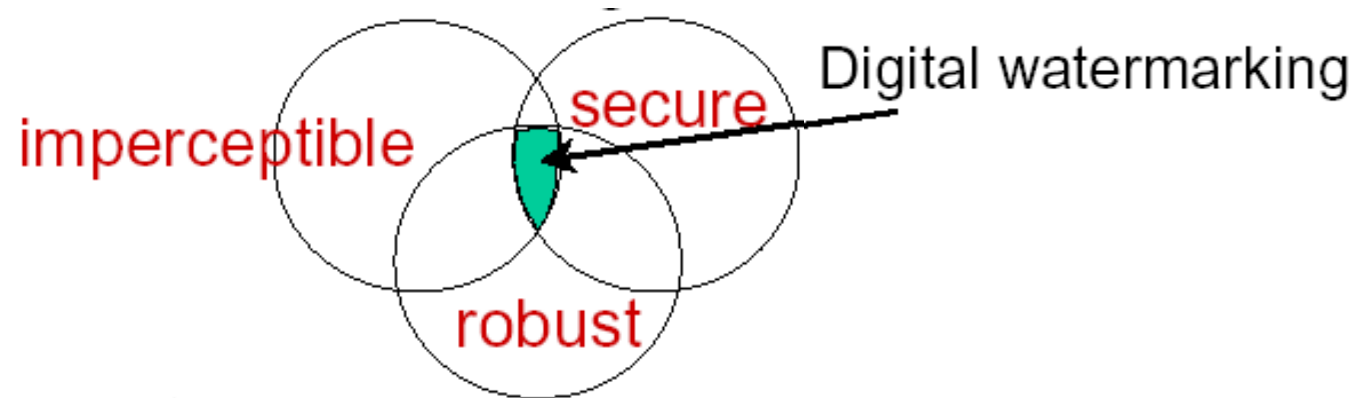


- *Watermark* melekat di dalam citra
- Penyisipan *watermark* tidak merusak kualitas citra
- *Watermark* dapat dideteksi/ekstraksi kembali sebagai bukti kepemilikan/*copyright*



# Image Watermarking (3)

- Persyaratan umum:
  - *imperceptible*
  - *robustness*
  - *secure*



# Metode Konvensional Memberi Label *Copyright*



- Label *copyright* ditempelkan pada gambar.
- Kelemahan: tidak efektif melindungi *copyright* sebab label bisa dipotong atau dibuang dengan program pengolahan citra komersil (ex: *Adobe Photoshop*).





*Original image + label copyright*



*Cropped image*





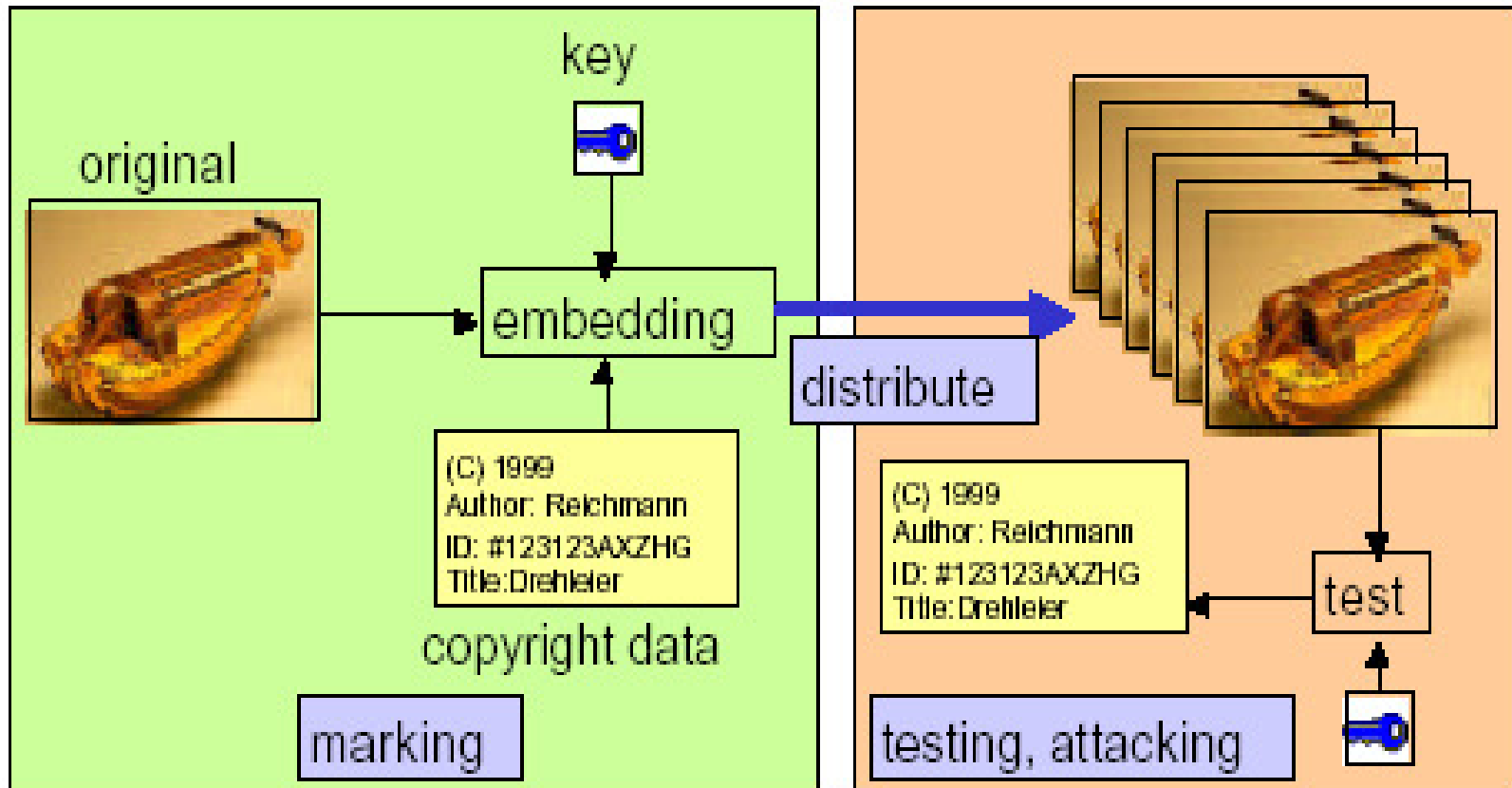
Muaro | photo by Lili Aini  
<http://liliaini.multiply.com>

Label kepemilikan



# Metode *watermarking*

- *Watermark* disisipkan ke dalam data multimedia.
- *Watermark* terintegrasi di dalam data multimedia.
- Kelebihan:
  1. Setiap penggandaan (*copy*) data multimedia akan membawa *watermark* di dalamnya.
  2. *Watermark* tidak bisa dihapus atau dibuang.



# Sejarah *Watermarking*



- Abad 13, pabrik kertas di Fabriano, Italia, membuat kertas yang diberi *watermark* dengan cara menekan bentuk cetakan gambar pada kertas yang baru setengah jadi.
- Ketika kertas dikeringkan terbentuklah suatu kertas yang ber-*watermark*. Kertas ini biasanya digunakan oleh seniman/sastrawan untuk menulis karya seni.
- Kertas yang sudah dibubuhi tanda-air dijadikan identifikasi bahwa karya seni di atasnya adalah milik mereka.



# Perbedaan Steganografi dan *Watermarking*

## Steganografi:

- Tujuan: mengirim pesan rahasia apapun tanpa menimbulkan kecurigaan
- Persyaratan: aman, sulit dideteksi, sebanyak mungkin menampung pesan (*large capacity*)
- Komunikasi: *point-to-point*
- Media penampung tidak punya arti apa-apa (*meaningless*)



## ***Watermarking:***

- Tujuan: perlindungan *copyright*, pembuktian kepemilikan (*ownership*), *fingerprinting*
- Persyaratan: *robustness*, sulit dihapus (*remove*)
- Komunikasi: *one-to-many*
- Komentar lain: media penampung justru yang diberi proteksi, *watermark* tidak rahasia, tidak mementingkan kapasitas *watermark*

# Jenis-jenis *Watermarking*



- *Fragile watermarking*

Tujuan: untuk menjaga integritas/orisinalitas media digital.

- *Robust watermarking*

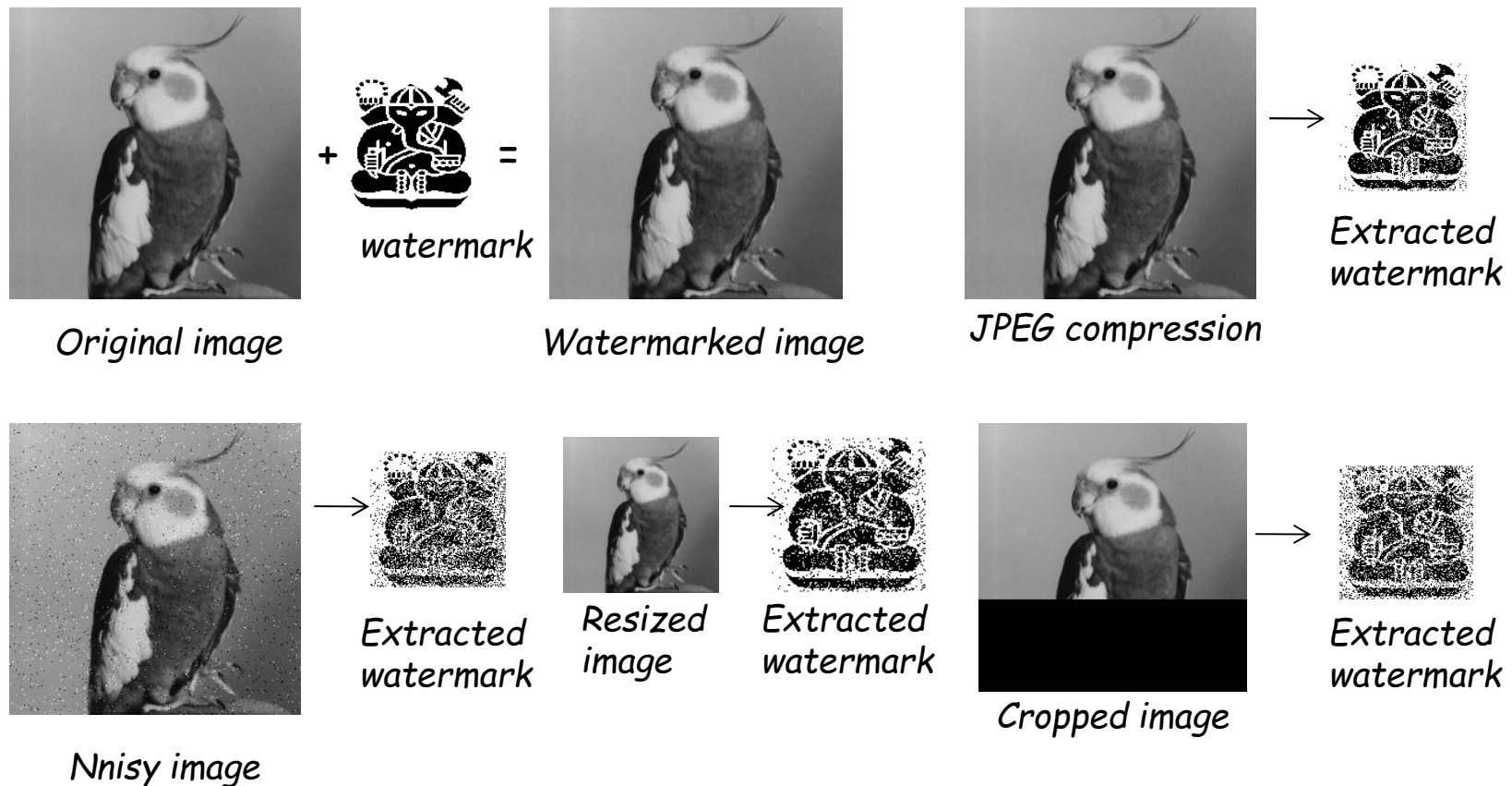
Tujuan: untuk menyisipkan label kepemilikan media digital.



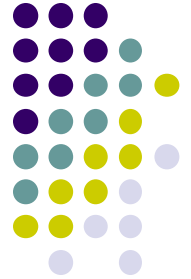
# Robust Watermarking







- *Watermark* tetap kokoh (*robust*) terhadap manipulasi (*common digital processing*) yang dilakukan pada media. Contoh: kompresi, *cropping*, *editing*, *resizing*, dll
- Tujuan: perlindungan *copyright*



# Robustness



<p>Citra asli</p> 	<p>Citra ber-<i>watermark</i></p> 
<p>Citra ber-<i>watermark</i> dikompresi 75%</p> 	<p>Citra ber-<i>watermark</i> di-<i>crop</i></p> 

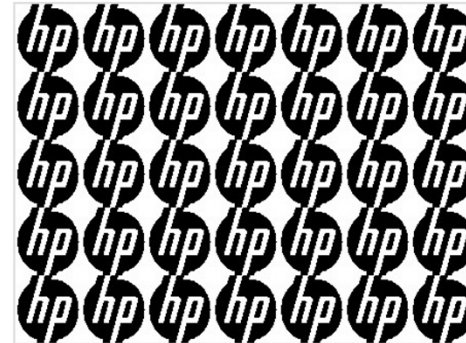
# Fragile Watermarking



- *Watermark* rusak atau berubah terhadap manipulasi (*common digital processing*) yang dilakukan pada media.
- Tujuan: pembuktian keaslian dan *tamper proofing*



(a)



(b)

Penambahan noise



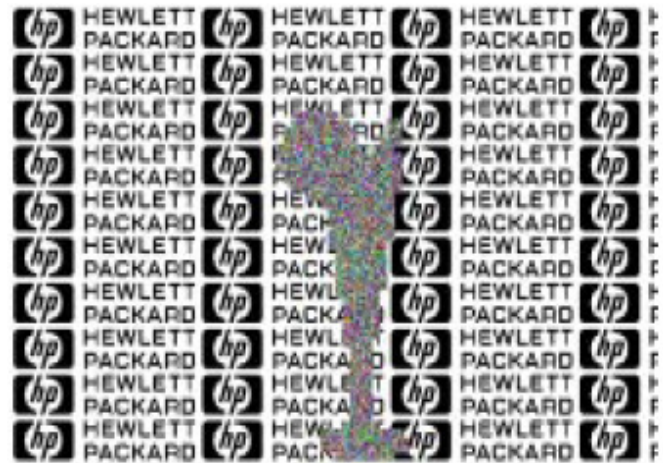
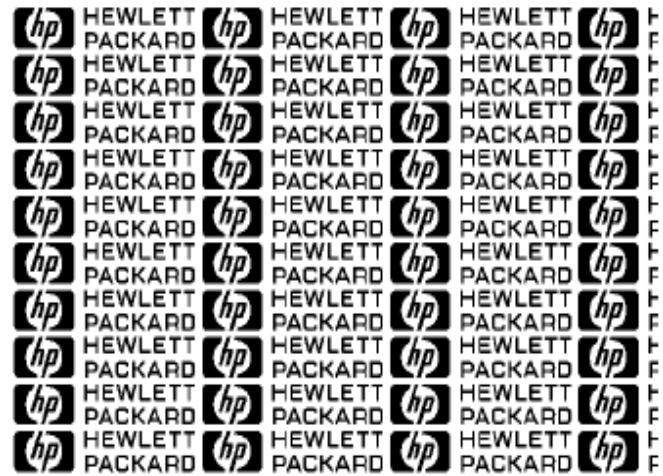
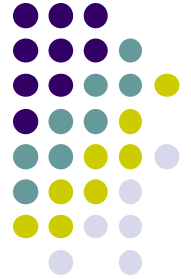
(c)



(d)

Watermark rusak



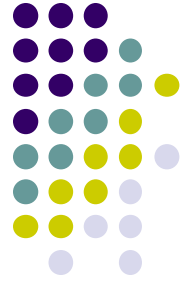


Contoh *fragile watermarking* lainnya (Wong, 1997)

# Data apa saja yang bisa *di-watermark*?



- Citra → *Image Watermarking*
- Video → *Video Watermarking*
- Audio → *Audio Watermarking*
- Teks → *Text Watermarking*
- Perangkat lunak → *Software watermarking*



# ***Watermarking pada Citra***

- *Visible Watermarking*
- *Invisible Watermarking*

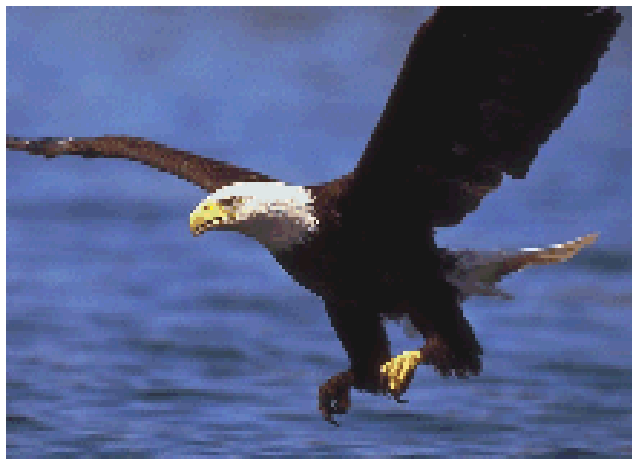


# *Visible Watermarking*

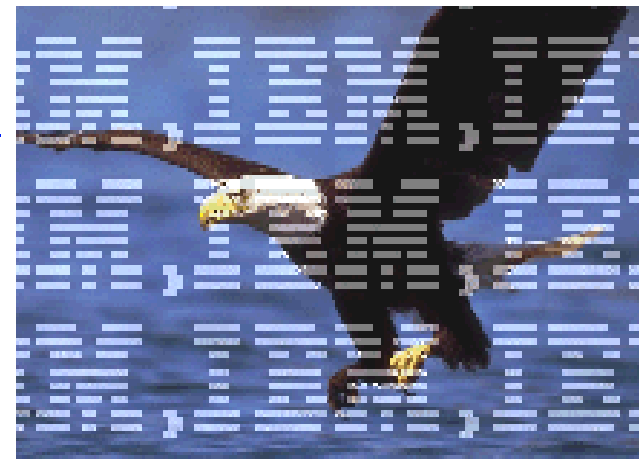
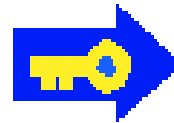




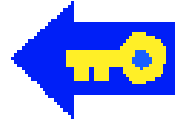
# *Visible Watermarking*



Embed



Remove







# ***Invisible Watermarking***

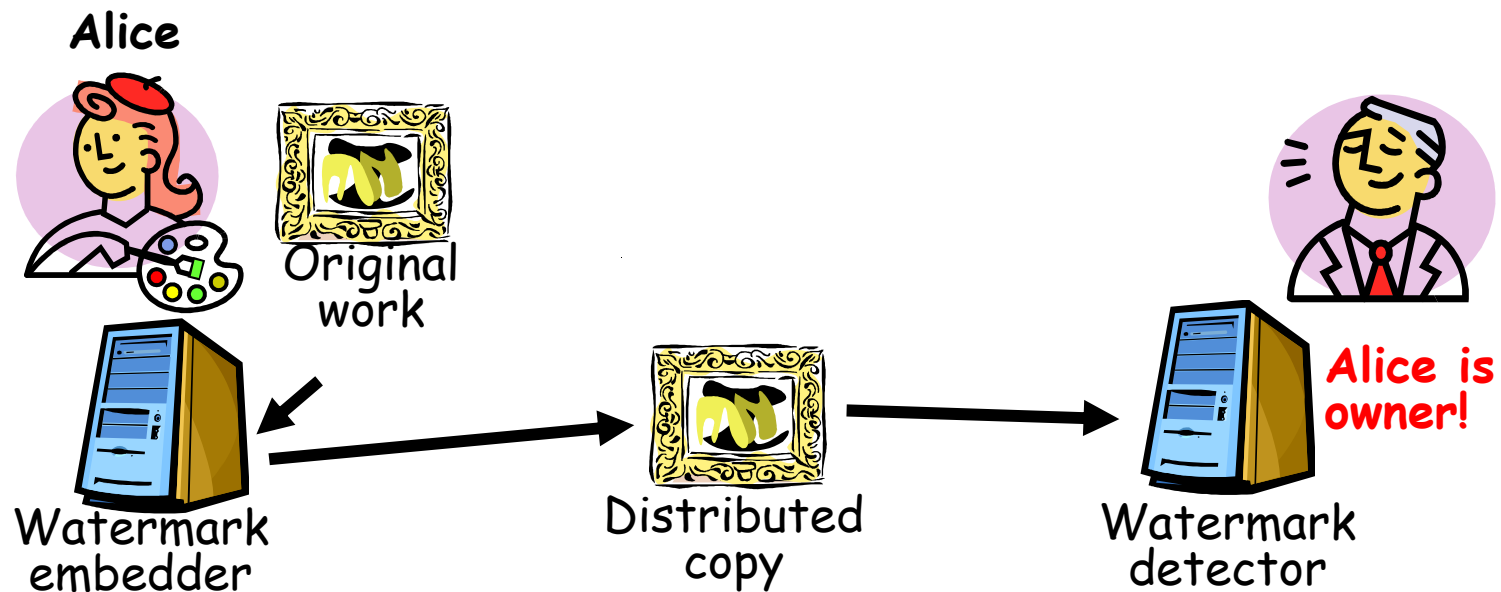




# Aplikasi *Watermark*

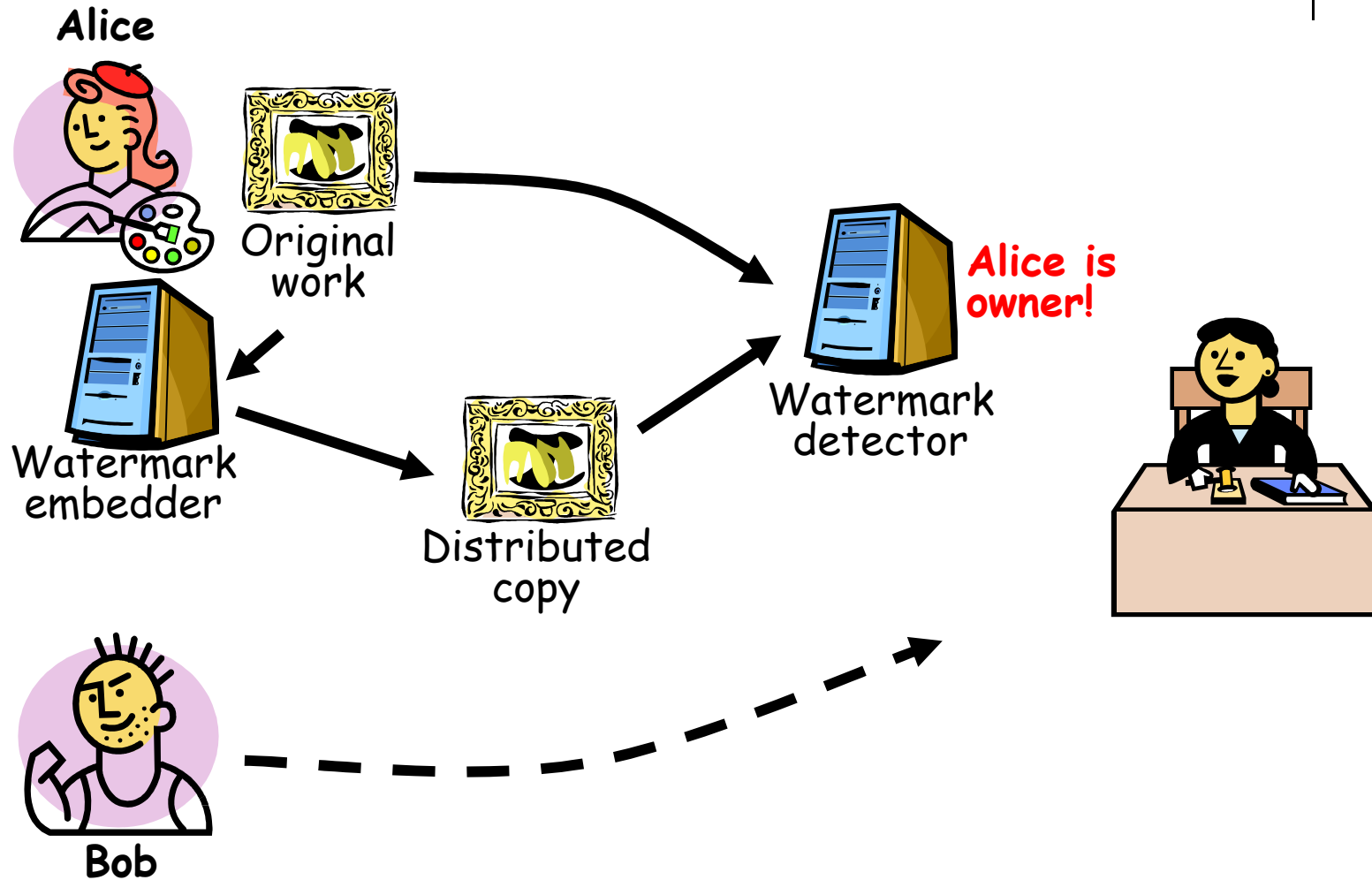
- Identifikasi kepemilikan (*ownership identification*)
- Bukti kepemilikan (*proof of ownership*)
- Memeriksa keaslian isi karya digital (*tamper proofing*) → *Content authentication*
- *User authentication/fingerprinting/transaction tracking*: mengotentikasi pengguna spesifik.  
Contoh: distribusi DVD
- *Piracy protection/copy control*: mencegah penggandaan yang tidak berizin.
- Broadcast monitoring

# Owner identification



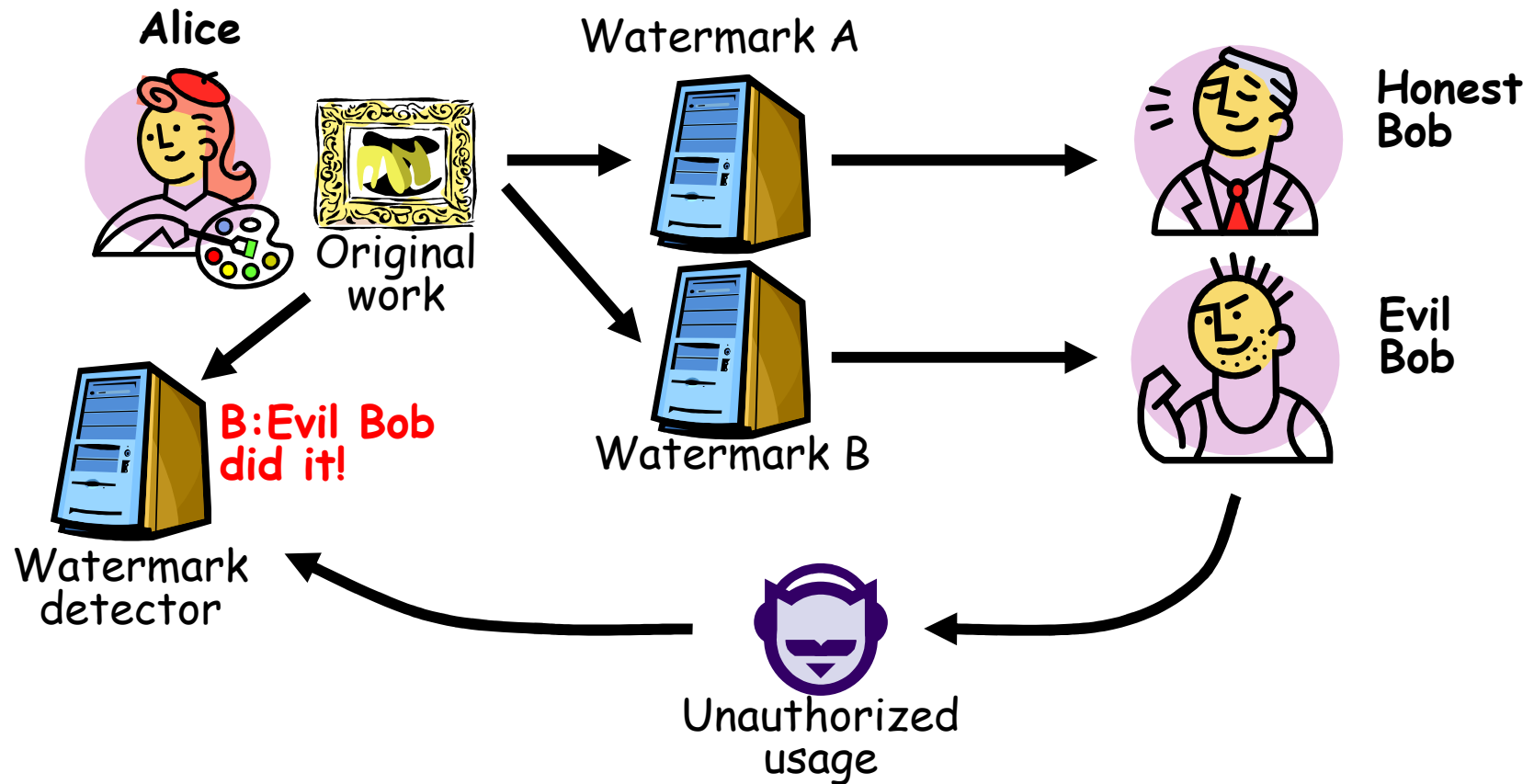


# Proof of ownership



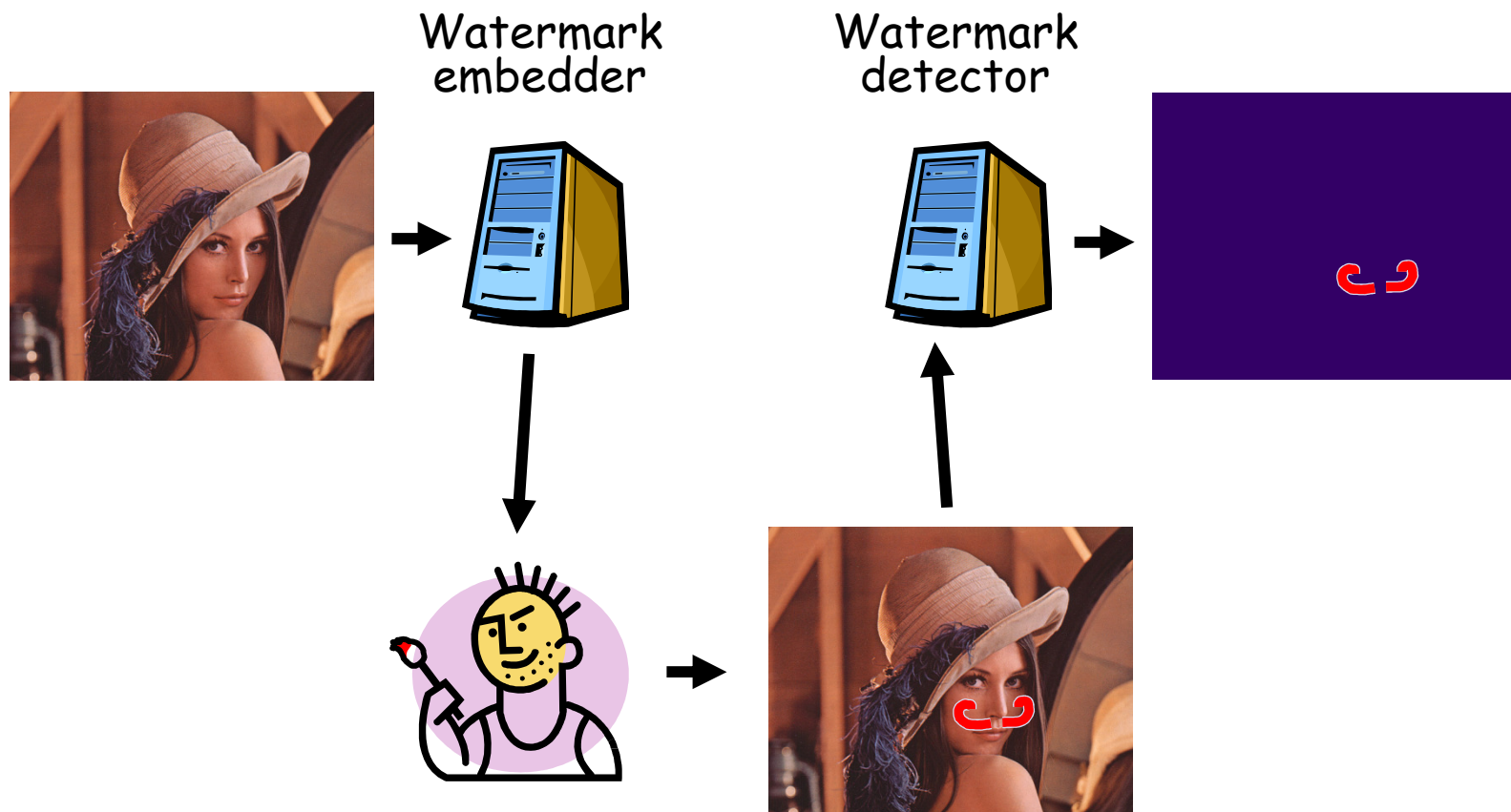


# Transaction tracking





# Content authentication



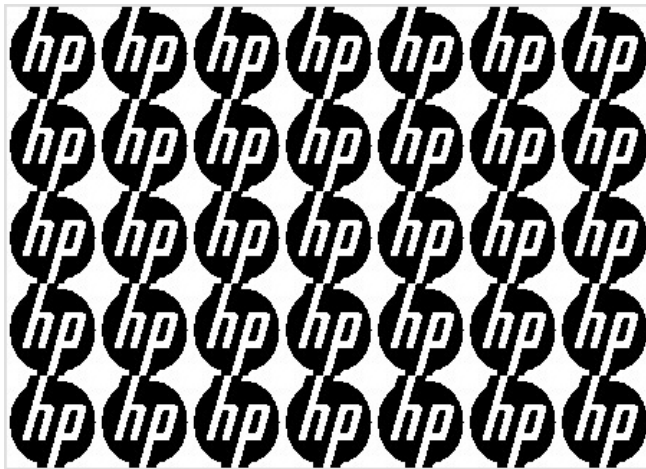




(a)



(b)



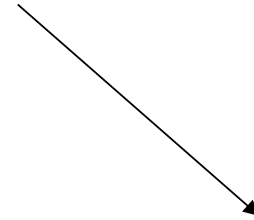
(c)



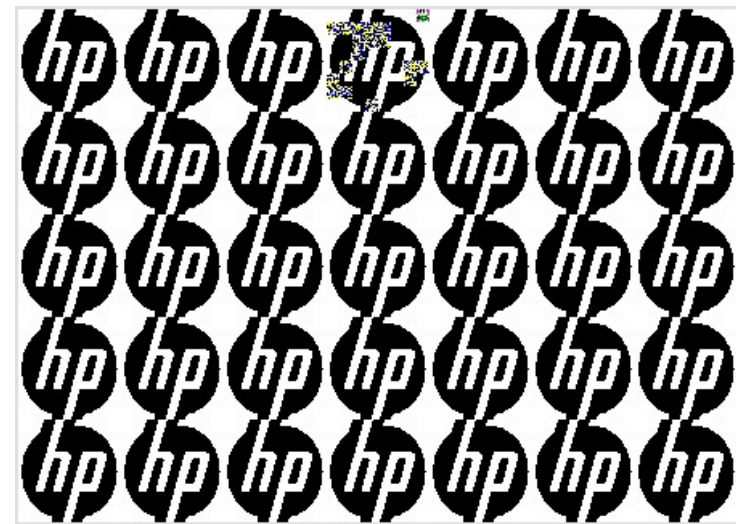
(d)



Watermark rusak



(e)



(f)

Kesimpulan: citra sudah mengalami modifikasi





# Content authentication



Original



Hasil pengubahan



(a) Clinton and Monica

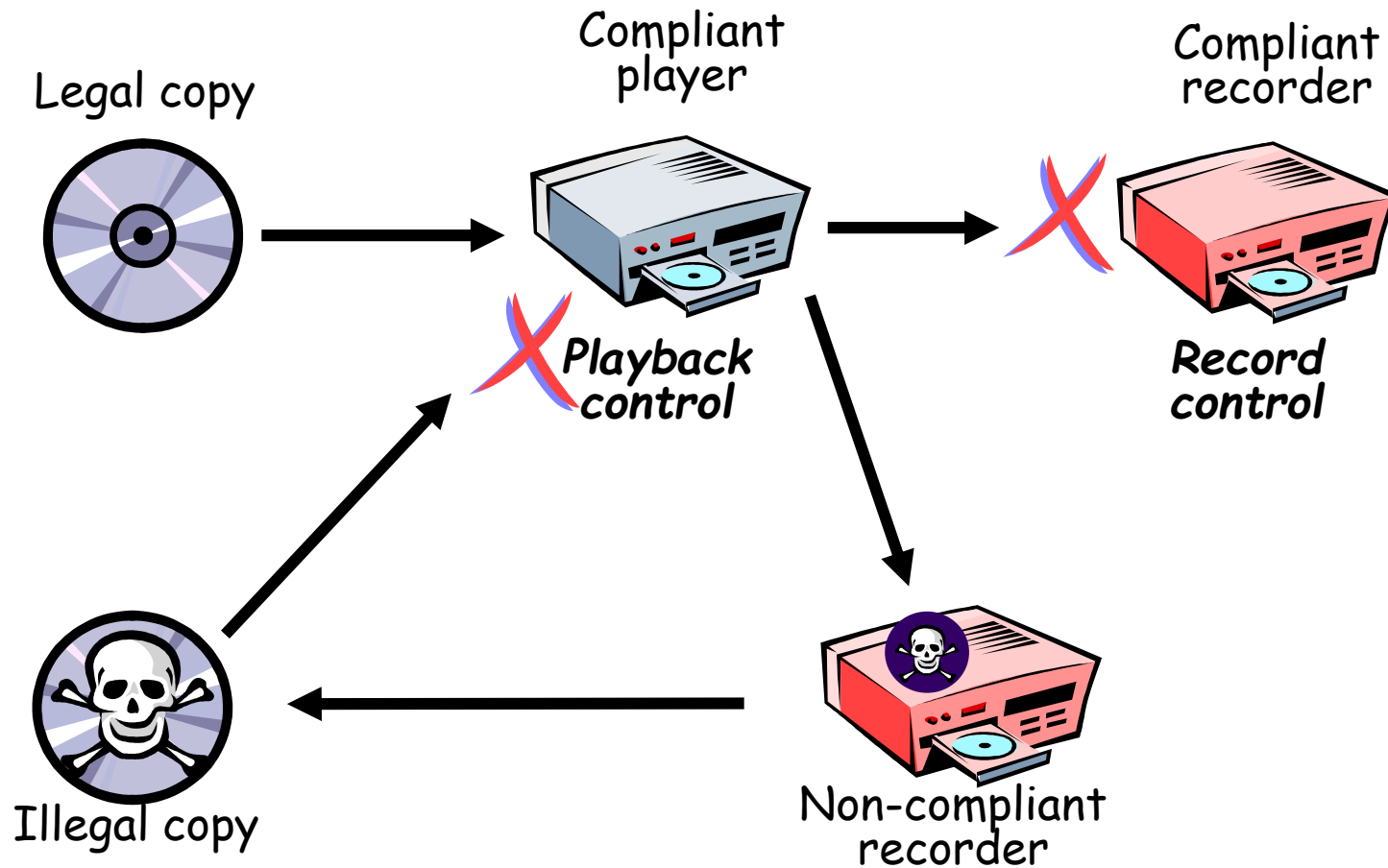
Foto mana yang asli?



(b) Clinton and Hillary

## 5. Copy control/Piracy Control

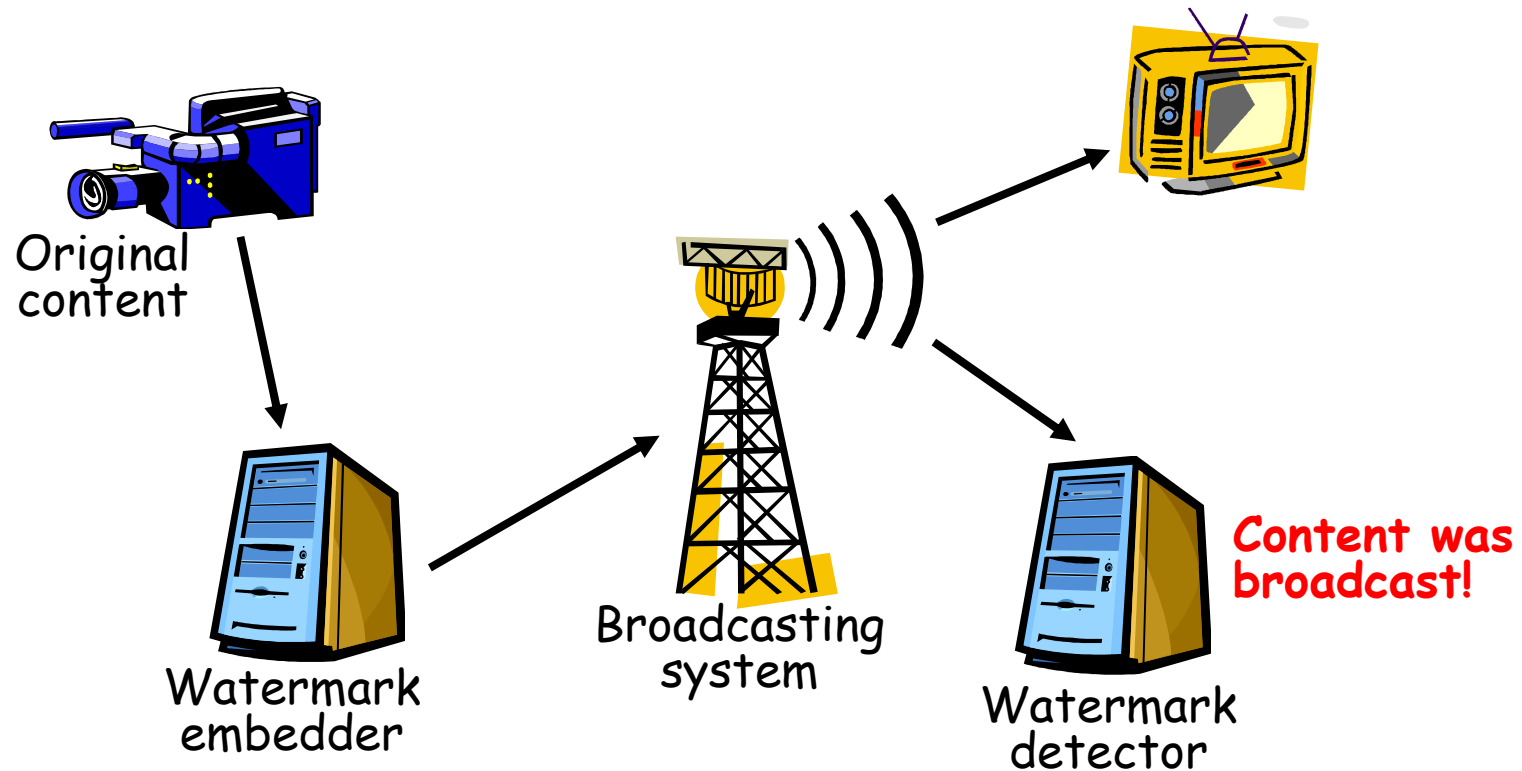
Watermark digunakan untuk mendeteksi apakah media digital dapat digandakan (copy) atau dimainkan oleh perangkat keras.





## 6. Broadcast monitoring

Watermark digunakan untuk memantau kapan konten digital ditransmisikan melalui saluran penyiaran seperti TV dan radio.





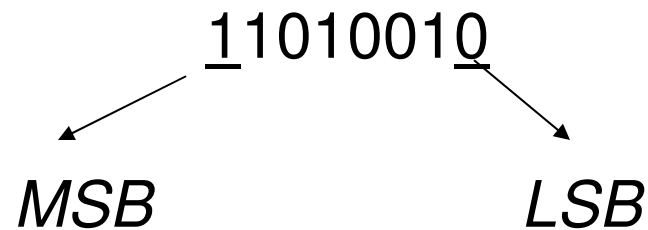
# Metode *Image Watermarking*

- Metode dalam ranah spasial  
Menyisipkan *watermark* langsung pada nilai *byte* dari *pixel* citra.
- Metode dalam ranah *transform*  
Menyisipkan watermark pada koefisien transformasi dari citra.



# Metode LSB

- Sama seperti steganografi.
- Mengganti bit *LSB* dengan bit data.



*LSB = Least Significant Bit*

*MSB = Most Significant Bit*

- Mengubah bit *LSB* hanya mengubah nilai *byte* satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya



- Misalkan sebagian pixel adalah citra

00110011    10100010    11100010    01101111  
(sekelompok *pixel* berwarna merah)

- Misalkan *watermark*: 0111

- *Encoding*:

00110010    10100011    11100011    01101110  
(*pixel* berwarna “merah berubah sedikit”)



- Kelemahan:
  1. tidak kokoh terhadap perubahan
  2. mudah dihapus dengan mengganti semua bit *LSB* dari media ber-*watermark*.



# Metode *Spread Spectrum*

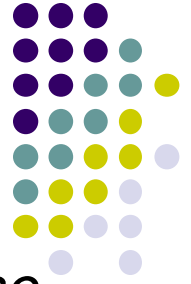


- Diusulkan pertama kali oleh Cox dalam makalah “*Secure Spread Spectrum Watermarking for Multimedia*” (1997)
- *Watermark* disebar (*spread*) di dalam citra.
- *Spread spectrum* dapat dilakukan dalam 2 ranah:
  1. Ranah spasial  
Menyisipkan *watermark* langsung pada nilai *byte* dari *pixel* citra.
  2. Ranah *transform*  
Menyisipkan *watermark* pada koefisien transformasi dari citra.

# Spread Spectrum



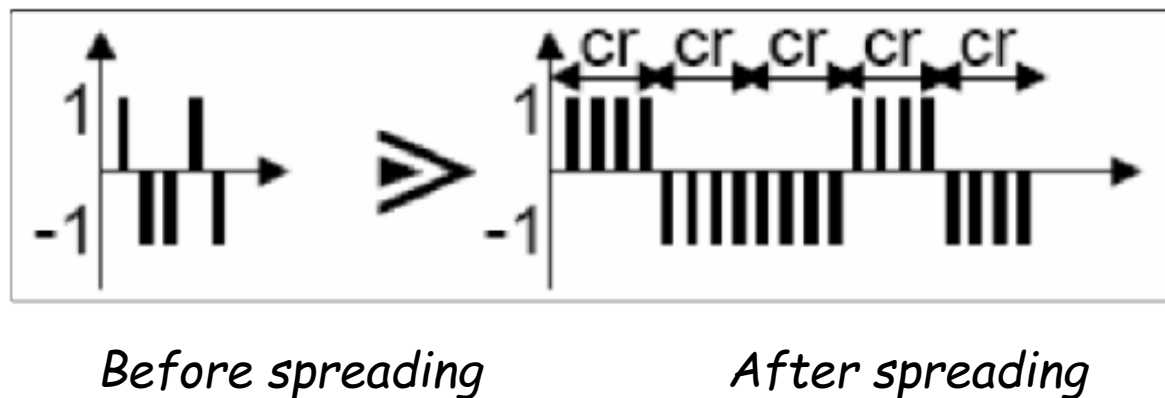
- Merupakan bentuk komunikasi menggunakan frekuensi radio.
- Tujuannya untuk menyembunyikan informasi di dalam kanal frekuensi radio yang lebar sehingga informasi akan tampak seperti *noise*.
- Teknik *spread spectrum* mentransmisikan sinyal informasi pita-sempit (*narrow band signal information*) ke dalam sebuah kanal pita lebar dengan penyebaran frekuensi.
- Artinya, data yang dikirim dengan metode *spread spectrum* menyebar pada frekuensi yang lebar.

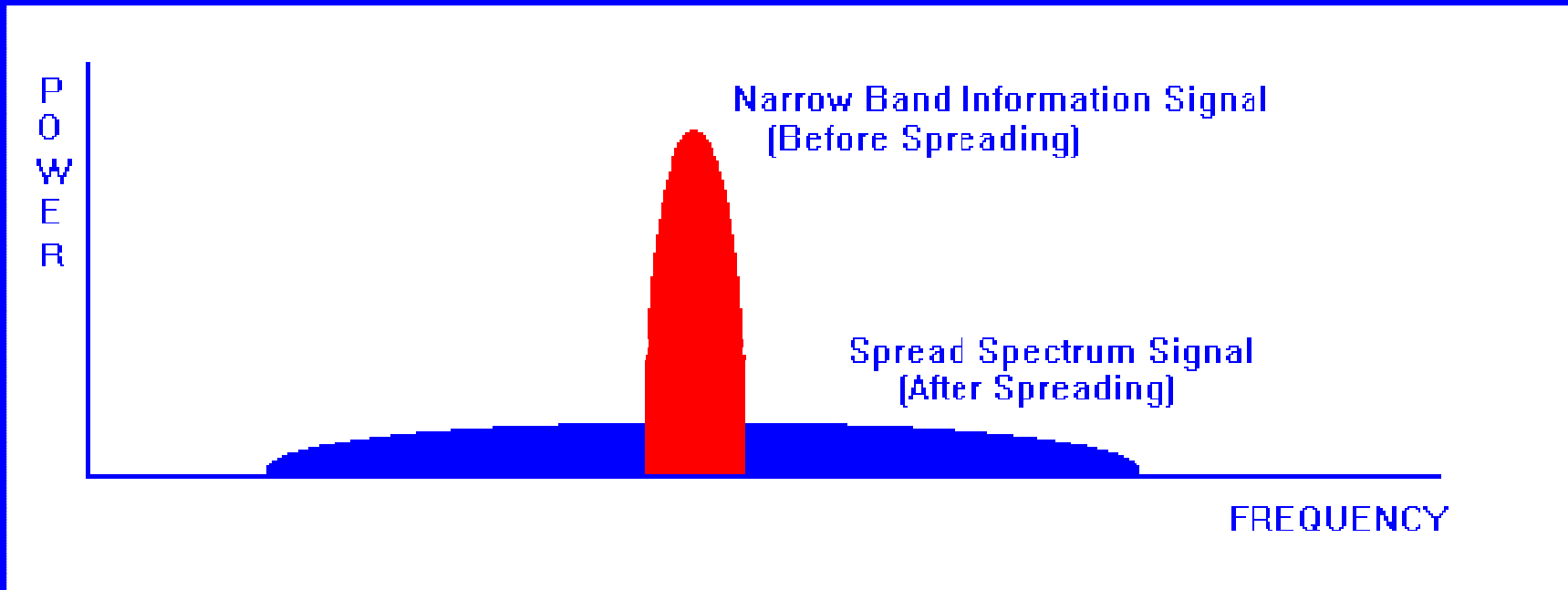
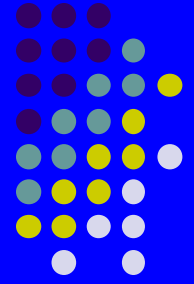


- Data yang disebar tampak terlihat seperti sinyal *noise* (*noise like signal*) sehingga sulit dideteksi, dicari, atau dimanipulasi.
- Metode *spread spectrum* awalnya digunakan di dalam militer, karena teknologi *spread spectrum* memiliki kemampuan istimewa sbb:
  1. Menyelundupkan informasi
  2. Mengacak data.
- Teknik *spread spectrum* ditemukan pada tahun 1930, dan dipatenkan pada tahun 1941 oleh Hedy Lamar dan George Antheil - *secret communications system used by the military.*



- Penyebaran data berguna untuk menambah tingkat redundansi.
- Besaran redundansi ditentukan oleh faktor pengali  $cr$  (singkatan dari *chip-rate*)
- Panjang bit-bit hasil redundansi menjadi  $cr$  kali panjang bit-bit awal.

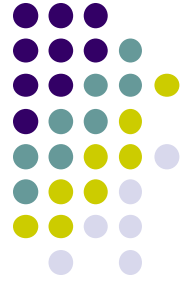






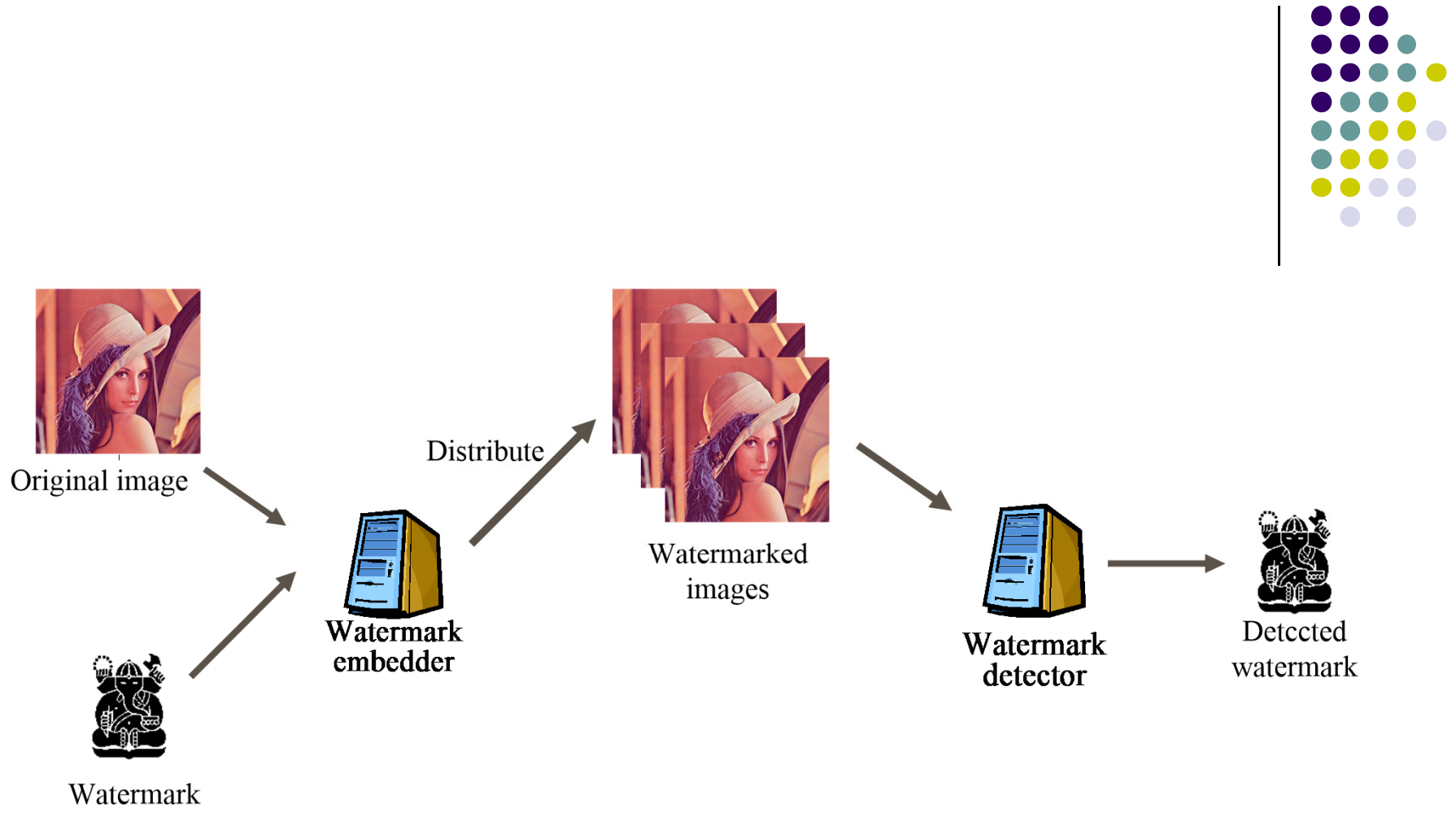
- Contoh: pesan = 10110 dan  $cr = 4$ , maka hasil *spreading* adalah 11110000111111110000
- Dengan teknik *spread spectrum*, data (pesan) dapat ditransmisikan tanpa penyembunyian tambahan, karena sudah tersembunyi dengan sendirinya.
- Ide *spread spectrum* ini digunakan di dalam *watermarking* adalah untuk memberikan tambahan keamanan pada pesan dengan cara menempelkan pesan di dalam media lain seperti gambar, musik, video, atau artikel (teks).

# Spread Spectrum Watermarking



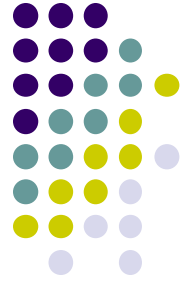
- Pesan, yang disebut *watermark*, disisipkan ke dalam media dalam ranah frekuensi. Umumnya *watermark* berupa citra biner seperti logo atau penggalan musik.
- Penyisipan dalam ranah frekuensi membuat *watermark* lebih kokoh (*robust*) terhadap serangan (*signal processing*) ketimbang penyisipan dalam ranah spasial (seperti metode modifikasi LSB) → *robust watermarking*.







- Misalkan media yang akan disisipi pesan (watermark) adalah citra (image).
- Terlebih dahulu citra ditransformasi dari ranah spasial ke dalam ranah frekuensi.
- Kakas transformasi yang digunakan antara lain:
  1. *Discrete Cosine Transform* (DCT)
  2. *Fast Fourier Transform* (FFT)
  3. *Discrete Wavelet Transform* (DWT)
  4. *Fourier-Mellin Transform* (FMT)



- *Discrete Cosine Transform (DCT)*

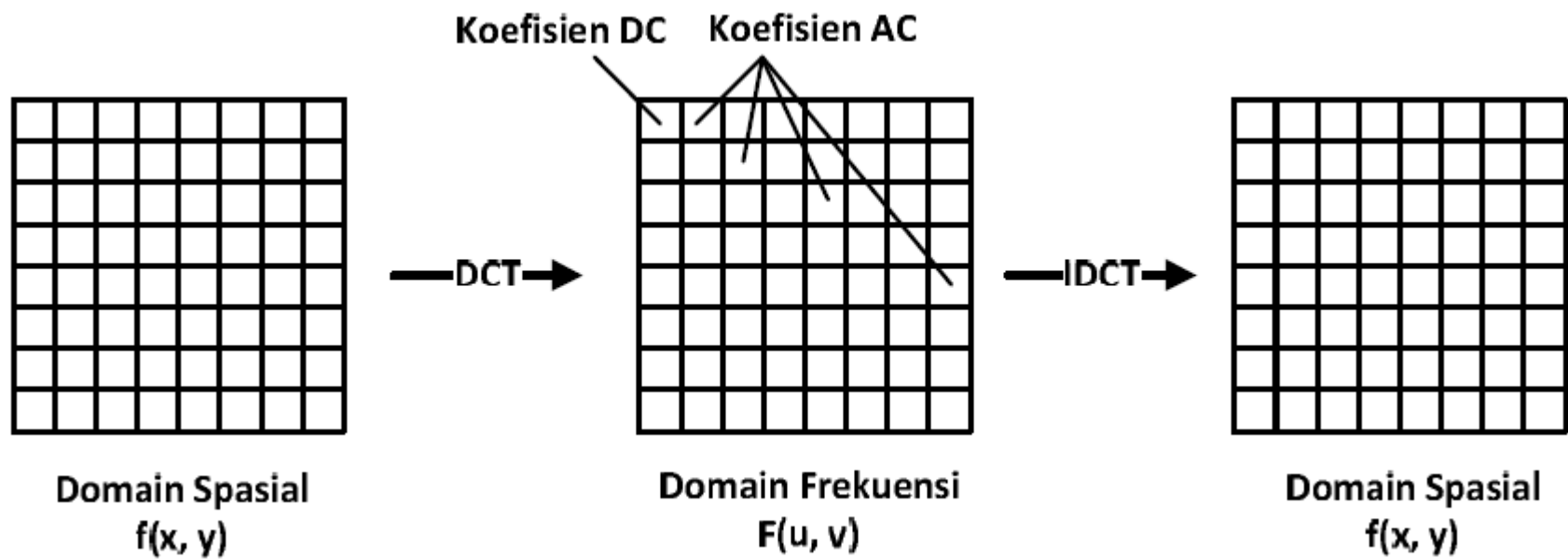
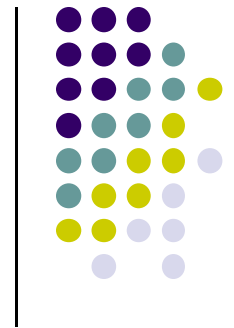
$$C(u, v) = \alpha_u \alpha_v \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I(x, y) \cos \frac{\pi(2x+1)u}{2M} \cos \frac{\pi(2y+1)v}{2N} \quad (1)$$

$$\alpha_u = \begin{cases} \frac{1}{\sqrt{M}} & , u = 0 \\ \sqrt{\frac{2}{M}} & , 1 \leq u \leq M-1 \end{cases} \quad \alpha_v = \begin{cases} \frac{1}{\sqrt{N}} & , v = 0 \\ \sqrt{\frac{2}{N}} & , 1 \leq v \leq N-1 \end{cases}$$

*C(u,v) disebut koefisien-koefisien DCT*

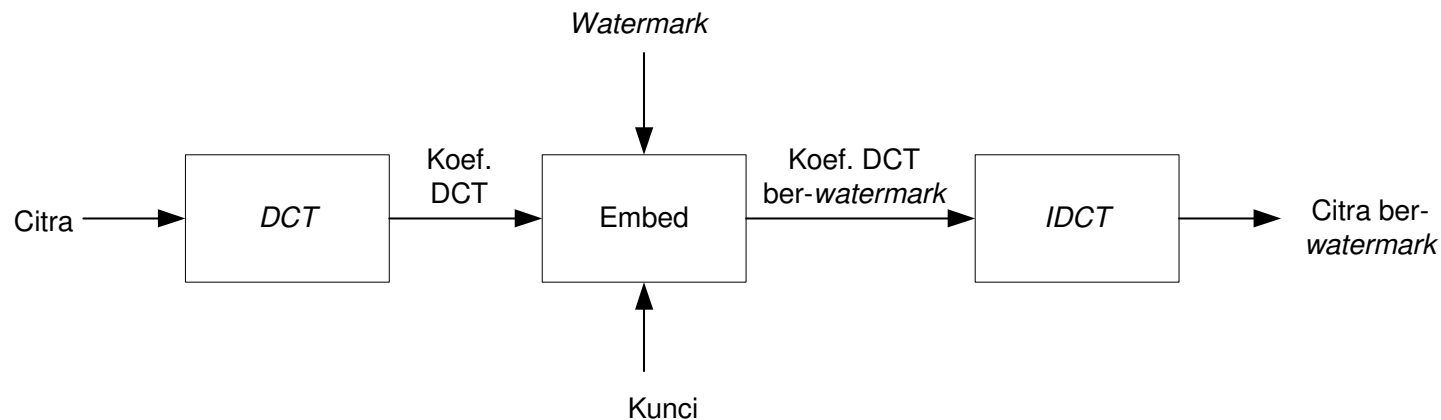
- *Inverse Discrete Cosine Transform (IDCT)*

$$I(x, y) = \alpha_u \alpha_v \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} C(u, v) \cos \frac{\pi(2x+1)u}{2M} \cos \frac{\pi(2y+1)v}{2N} \quad (4)$$



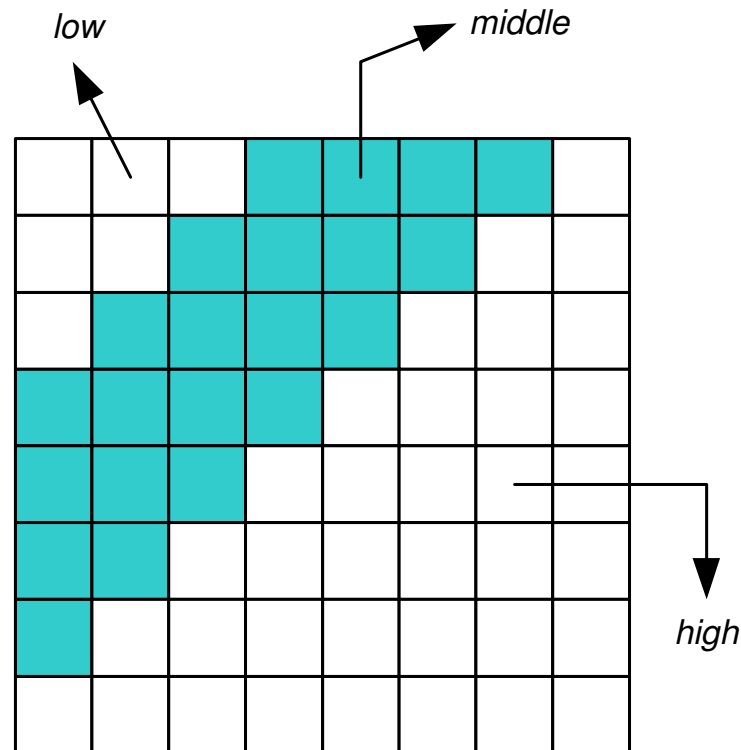


- Hasil tranformasi menghasilkan nilai-nilai yang disebut koefisien-koefisien transformasi (misalnya koefisien DCT).
- Bit-bit pesan disembunyikan pada koefisien-koefisien tranformasi tersebut (dengan suatu formula atau disisipkan pada bit LSB koefisien transformasi).
- Selanjutnya, citra ditransformasikan kembali (*inverse transformation*) ke ranah spasial untuk mendapatkan citra stegano (atau citra *ber-watermark*).





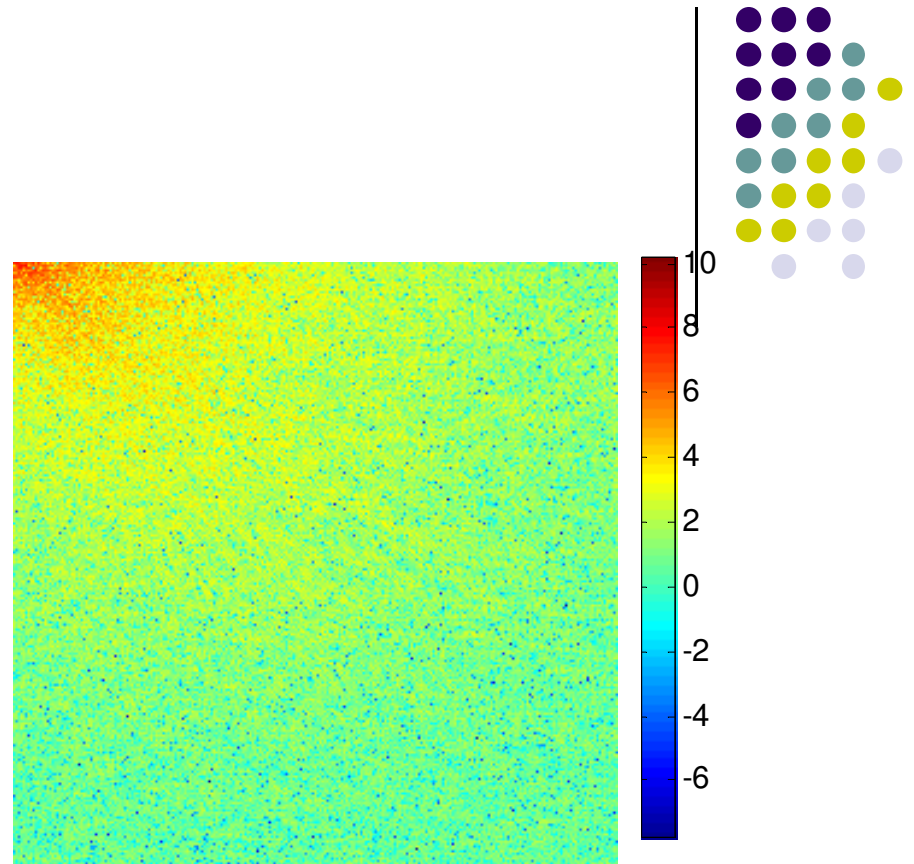
- Tinjau kakas transformasi yang digunakan adalah DCT.
- DCT membagi citra ke dalam 3 ranah frekuensi: *low frequencies*, *middle frequencies*, dan *high frequencies*)







Citra dalam ranah spasial



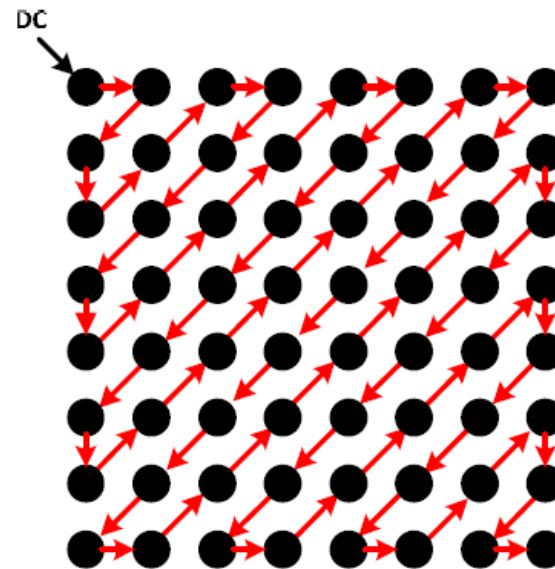
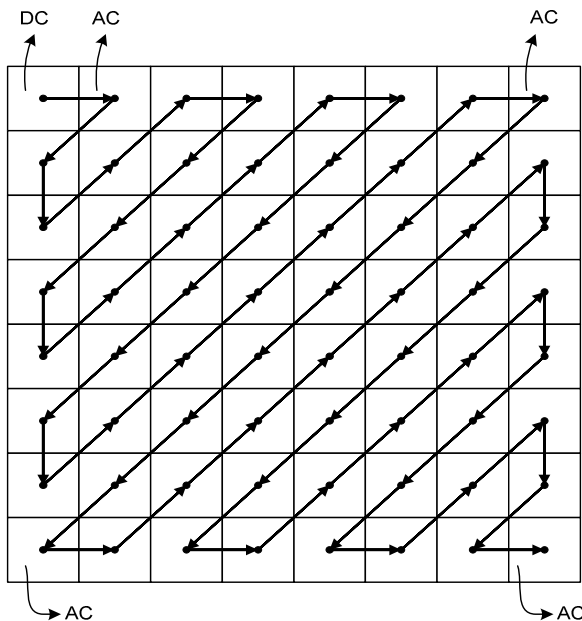
Citra dalam ranah frekuensi



- Bagian *low frequency* berkaitan dengan tepi-tepi (*edge*) pada citra, sedangkan bagian *high frequency* berkaitan dengan detail pada citra.
- Penyisipan pada bagian *low frequency* dapat merusak citra karena mata manusia lebih peka pada frekuensi yang lebih rendah daripada frekuensi lebih tinggi.
- Sebaliknya bila *watermark* disisipkan pada bagian *high frequency*, maka *watermark* tersebut dapat terhapus oleh operasi kuantisasi seperti pada kompresi *lossy* (misalnya *JPEG*).



- Oleh karena itu, untuk menyeimbangkan antara *robustness* dan *imperceptibility*, maka *watermark* disisipkan pada bagian *middle frequency* (bagian yang diarsir pada Gambar di atas).
- Bagian *middle frequency* diekstraksi dengan cara membaca matriks *DCT* secara *zig-zag* sebagaimana yang dilakukan di dalam algoritma kompresi *JPEG*



Pembacaan secara zigzag

# Skema Penyisipan

1. Misalkan

$$A = \{a_i \mid a_i \in \{-1, +1\}\} \quad (1)$$

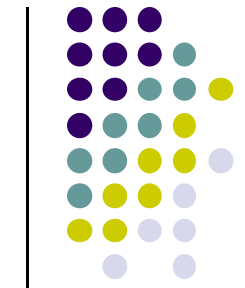
adalah bit-bit pesan (*watermark*) yang akan disembunyikan di dalam citra (catatan: bit 1 dinyatakan sebagai +1 dan bit 0 sebagai -1)

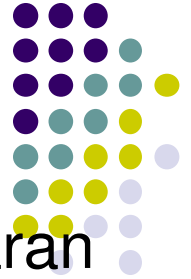
2. Setiap bit  $a_j$  dilakukan *spreading* dengan faktor  $cr$  yang besar, yang disebut *chip-rate*, untuk menghasilkan barisan:

$$B = \{b_i \mid b_i = a_j, j \cdot cr \leq i < (j + 1) \cdot cr\}. \quad (2)$$

3. Bit-bit hasil *spreading* kemudian dimodulasi dengan barisan bit acak (*pseudo-noise*):

$$P = \{p_i \mid p_i \in \{-1, 1\}\} \quad (3)$$





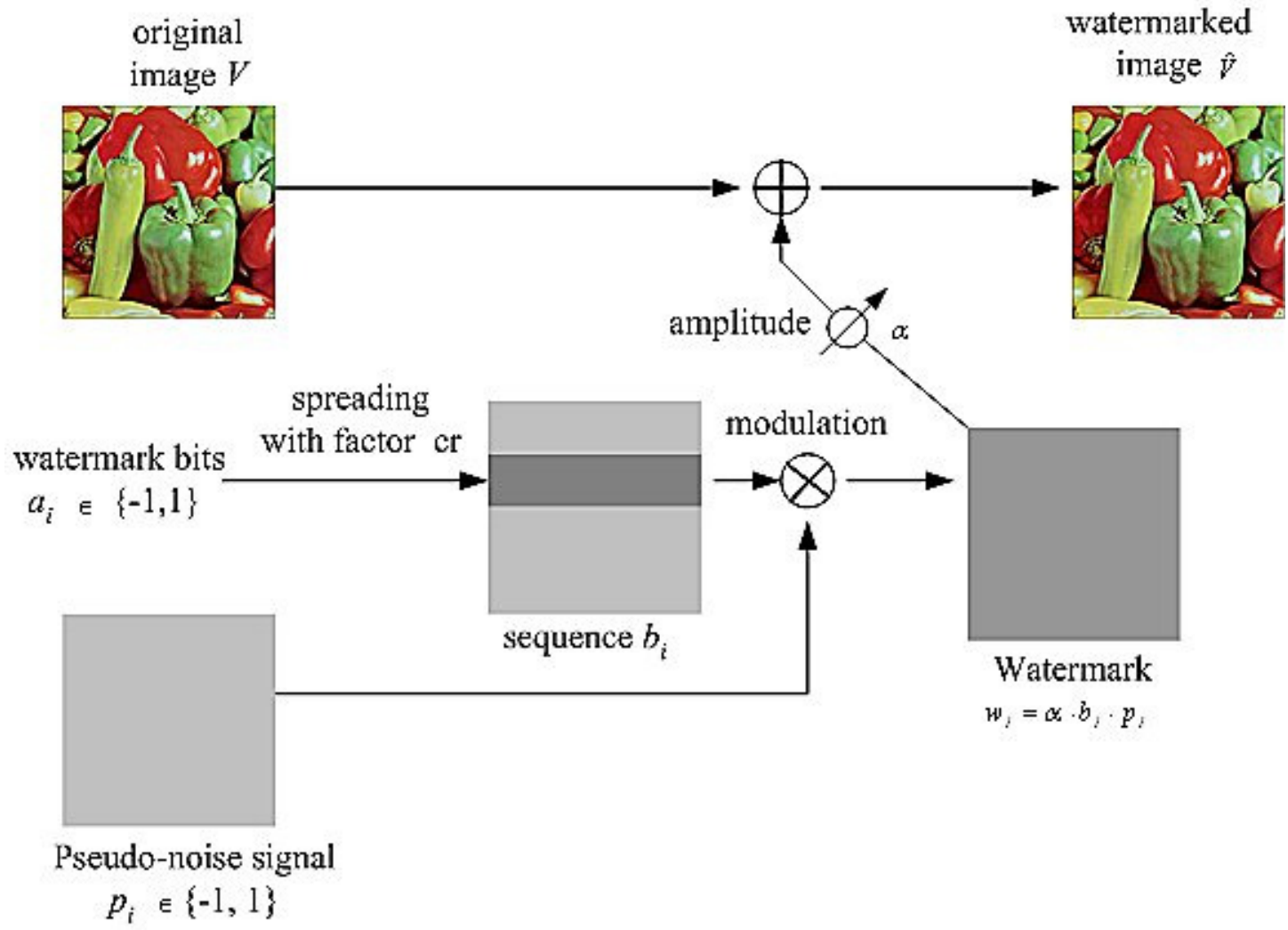
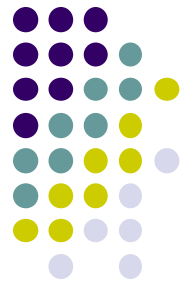
4. Bit-bit  $p_i$  diamplifikasi (diperkuat) dengan faktor kekuatan (strength) watermarking  $\alpha$  untuk membentuk *spread spectrum watermark*

$$w_i = \alpha \cdot b_i \cdot p_i \quad (4)$$

5. *Watermark*  $w_i$  disisipkan ke dalam citra (dalam ranah frekuensi)  $V = \{v_i\}$  dengan persamaan:

$$\hat{v}_i = v_i + w_i \quad (5)$$

Dikaitkan dengan sifat *noisy*  $p_i$ ,  $w_i$  juga *a noise-like signal* dan sulit dideteksi, dicari, dan dimanipulasi.







## Skema Ekstraksi

Untuk mengekstraksi pesan (*watermark*) dari citra stegano, penerima pesan harus memiliki *pseudo-noise*  $p_i$  yang sama dengan yang digunakan pada waktu penyisipan.

Ekstraksi pesan dilakukan dengan cara sebagai berikut:

1. Kalikan citra stegano dengan  $p_i$ :

$$\sum_{i=j \cdot cr}^{(j+1) \cdot cr - 1} p_i \cdot \hat{v}_i = \sum_{i=j \cdot cr}^{(j+1) \cdot cr - 1} v_i \cdot p_i \quad (6)$$
$$+ \sum_{i=j \cdot cr}^{(j+1) \cdot cr - 1} \alpha \cdot b_i \cdot p_i^2$$



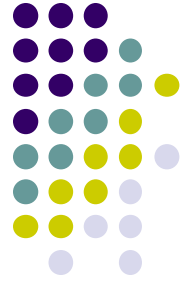
Karena  $p_i$  acak,  $cr$  besar, dan deviasi  $v_i$  kecil, maka dapat diharapkan bahwa

$$\lim_{cr \rightarrow \infty} \sum_{i=j \cdot cr}^{(j+1) \cdot cr - 1} v_i \cdot p_i \approx 0 \quad (7)$$

Karena  $p_i^2 = 1$ , persamaan (6) menghasilkan jumlah korelasi:

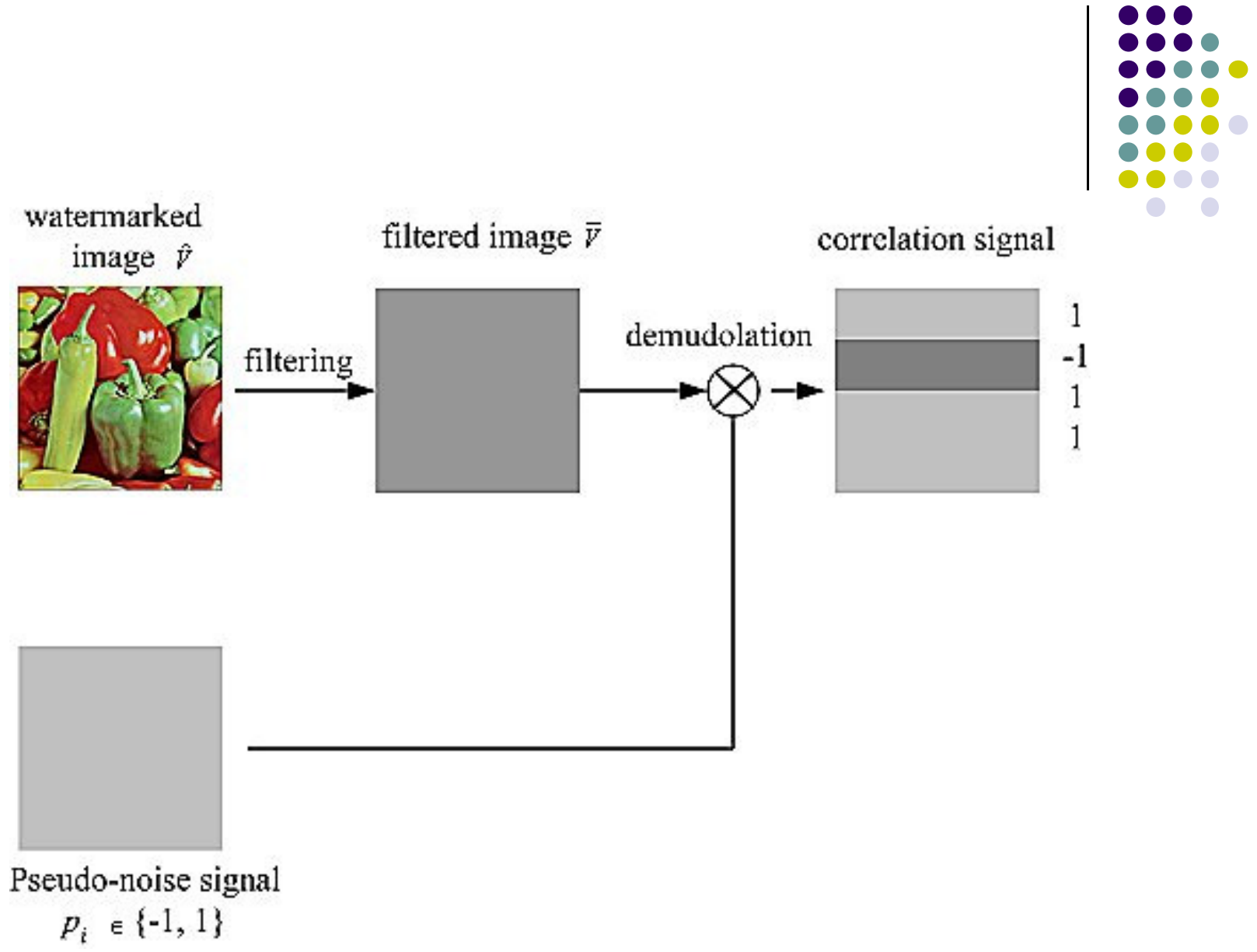
$$\sum_{i=j \cdot cr}^{(j+1) \cdot cr - 1} \hat{v}_i \cdot p_i = \alpha \cdot cr \cdot a_j \quad (8)$$

Oleh karena itu, bit-bit yang disisipkan dapat ditemukan kembali dengan langkah 2 berikut:



2. Bit-bit pesan diperoleh kembali dengan persamaan berikut:

$$a_j = \begin{cases} 1, & \sum_{i=j \cdot cr}^{(j+1) \cdot cr - 1} \hat{v}_i \cdot p_i > 0 \\ -1, & \sum_{i=j \cdot cr}^{(j+1) \cdot cr - 1} \hat{v}_i \cdot p_i < 0 \end{cases} \quad (9)$$





- Untuk membantu tahap korelasi, citra stegano dapat ditapis (*filtering*) terlebih dahulu dengan penapis lolos-tinggi seperti penapis Wiener atau penapis deteksi tepi.
- Penapisan dapat menghilangkan komponen yang timbul dari superposisi citra dan pesan (*watermark*).

# Algoritma *Spread Spectrum Steganography*



## A. Penyisipan pesan

1. Transformasi citra ke ranah frekuensi dengan menggunakan DCT. Simpan semua koefisien DCT di dalam matriks  $M$ .
2. Baca matrik  $M$  dengan algoritma *zigzag* untuk memperoleh koefisien-koefisien DCT, simpan di dalam vektor  $V$ .
3. *Spreading* pesan  $A$  dengan faktor  $cr$  untuk memperoleh barisan  $B$  dengan menggunakan persamaan (2). Misalkan panjang  $B$  adalah  $m$ .





4. Bangkitkan barisan *pseudo-noise*  $P$  sepanjang  $m$ .
5. Kalikan  $p_i$  dan  $b_i$  dan  $\alpha$  dengan persamaan (4) untuk menghasilkan  $w_i$ .
6. Sisipkan  $w_i$  ke dalam elemen-elemen  $V$  dengan persamaan (5). Untuk menyeimbangkan tingkat *imperceptibility* dan *robustness*, lakukan penyisipan pada *middle frequencies*. *Middle frequencies* dapat dipilih dengan melakukan lompatan pada  $V$  sejauh  $L$ .
7. Terakhir, terapkan IDCT untuk memperoleh citra stegano (*watermarked image*).



## B. Ekstraksi pesan

1. Transformasi citra stegano ke ranah frekuensi dengan menggunakan DCT. Simpan semua koefisien DCT di dalam matriks  $M$ .
2. Baca matrik  $M$  dengan algoritma *zigzag* untuk memperoleh koefisien-koefisien DCT, simpan di dalam vektor  $V$ .
3. Bangkitkan barisan *pseudo-noise*  $P$  sepanjang  $m$ .
4. Kalikan  $p_i$  dan  $v_i$  dengan persamaan (6).
5. Dapatkan kembali bit-bit pesan (*watermark*) dengan persamaan (9).



**Catatan:** Pesan yang diekstraksi tidak selalu tepat sama dengan pesan yang disisipkan, alasannya adalah:

1. DCT adalah transformasi yang *lossy*. Artinya, ada perubahan bit yang timbul selama proses transformasi. DCT beroperasi pada bilangan real. Operasi bilangan real tidak eksak karena mengandung pembulatan (*round-off*).
2. Bergantung pada awal posisi *middle frequency* yang digunakan ( $L$ ). Posisi awal *middle frequency* hanya dapat diperkirakan dan tidak dapat ditentukan dengan pasti.



Original image



Stego-image

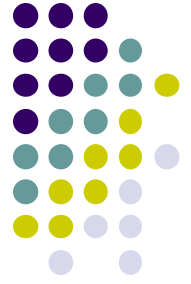


watermark



extracted watermark

# Referensi



1. Nick Sterling, Sarah Wahl, Sarah Summers, *Spread Spectrum Steganography*.
2. F. Hartung, and B. Girod, *Fast Public-Key Watermarking of Compressed Video*, Proceedings of the 1997 International Conference on Image Processing (ICIP '97).
3. Winda Winanti, *Penyembunyian Pesan pada Citra Terkompresi JPEG Menggunakan Metode Spread Spectrum*, Tugas Akhir Informatika ITB, 2009.