

1. Pembentukan Citra Digital

- a. Sampling dan Kuatisasi
- b. Penyimpanan Citra dalam File
- c. Halftoning

=====

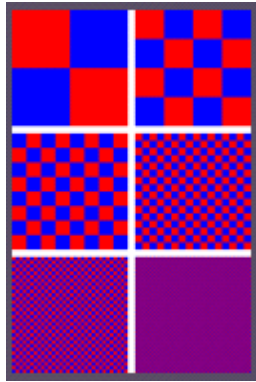
- a. Sampling dan Kuatisasi
 - i. Sampling
 - Digitaliasasi koordinat spatial
 - Merubah nilai kontinu menjadi nilai diskrit
 - ii. Kuantisasi
 - Digitalisasi amplitudo
 - Level dari warna/ gray level dalam setiap element array
 - iii. Jumlah bit dari suatu citre
 - 1) $\text{Bit} = N * M * q$ (color depth)
- b. Penyimpanan Citra dalam File (BMP)
 - i. Tiap file terdiri dari :
 - 1) File header : 14 bytes
 - 2) Bitmap header : 40 bytes
 - 3) Color map (kecuali 24-bit)
 - a) Berisi index warna dengan warna yang dipakai paling sering berada paling depan.
 - 4) Bitma data
 - ii. Konversi dari intel
 - 1) Low byte disimpan lebih dulu
 - iii. Penyimpanan pixel
 - 1) Pixel disimpan terbalik : bitmat data paling awal ada pada bagain gambar pling bawah (A) an paling akhir ada pada (B).



Screen clipping taken: 26/12/2011 10:32

- c. Halftoning
 - Merupakan metode untuk mencetak sejumlah besar rentang warna dengan perangkat yang terbatas, dengan cara menipu mata.
 - Contoh metode
 - Pixel Grid
 - ◆ Dengan $n \times n$ pixel pada sistem bilevel, maka akan didapat $n^2 + 1$ jumlah intensitas
- d. Dithering
 - Teknik untuk melakukan pendekatan halftoning dengan meminimalkan penurunan resolusi

- Misal : membuat warna aru dengan menggabungkan 2 warna.



- Beberapa teknik dithering :
 - Average dithering : memilih warna pixel yang paling dekat rata2 warna, sejenis pixel grid pada halftoning.
 - Dither noise : menambahkan noise random ke seluruh pixel
 - Error diffusion : error antara intensitas input dan intensitas pixel yang ditampilkan disebar (difusi) ke pixel2 di sebelah kan dan bawahnya.
 -
 -

2. Operasi2 dasar pengolahan citra

- Operasi aljabar
- Operasi aritmatike
 - Pengalian, penjumlahan, pengurangan, pembagian
- Operasi boolean
 - And, or, not, nand, nor, xor
- Operasi geometri
 - Zoom (in & out)
 - Rotasi flipping, cutpas, warping

3. Transformasi Fourier

- Dengan transformasi furier kita mengubah citra dari domain spasial ke domain frekuensi.
- Kuntungannya transformasi dapat dilakukan dengan operasi perkalian langsung.

4. Image Smoothing

- Dilakukan untuk menghilangkan efek pada citra digital yang s=disebabkan oleh keterbatasan sistem pencuplikan.
- Teknik penghalusan :
 - Domain spasial :
 - Mean
 - Median
 - Modus
 - Domain frekwensi :
 - Lowpass
 - Hight pass
- Efek samping, citra menjadi blur.
- =====
- Domain spasial :
 - Mean filtering :
 - Merata - ratakan nilai gray level dari pixel pada $f(x,y)$ yang termasuk kedalam area ketetangaan
 - Matrik mask :
 - ◆ Ukurannya sama dengan matrik ketanggan
 - ◆ Menyimpan operasi yang akan dikenakan pada matrik ketetanggan
 - ◆ Operasi secara skalar :

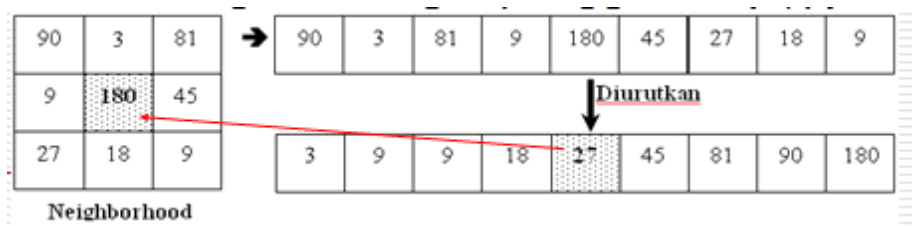
$$G(x,y) = \sum_{(n,m) \in S} Mask(n,m) \times Neighborhood(n,m)$$

Screen clipping taken: 26/12/2011 11:48

- Tingkat blur yang didapat sebanding dengan ukuran matrik yang digunakan.
- Penentuan kualitas smoothing dengan threshold.
- Masalah: penentuan threshold yang bagus memerlukan trial and error.

- Median Filtering

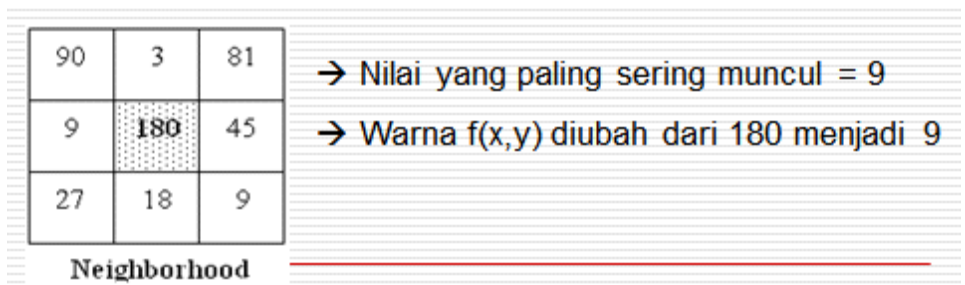
- Nilai media dari matrik ketetanggan digunakan menjadi nilai pada pixel f(x,y)
- Mencari median :



Screen clipping taken: 26/12/2011 11:53

- Median Filtering :

- Warna yang paling sering muncul dalam matrik ketetanggan digunakan sebagai warna pada f(x,y)
- Mencari modus :



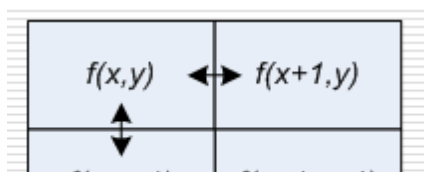
Screen clipping taken: 26/12/2011 11:55

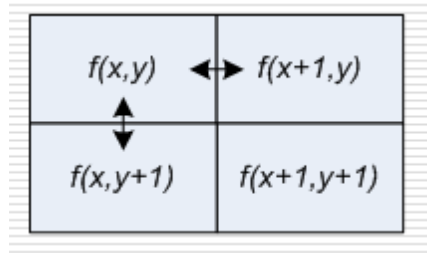
- Domain frekuensi :

- Low pass Filtering
- Bluring dapat dilakukan dengan menyaring / menghilangkan frekwensi tinggi pada citra.

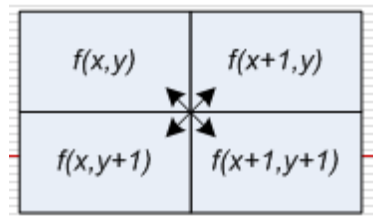
5. Image Sharpening

- a. Digunakan untuk memperjelas sisi pada citra
- b. Teknik sharpening :
 - i. Domain spatial :
 - Defferentiation
 - ◆ Averanging -> intergrasi ; sharpening -> turunan
 - ◆ Metode yang paling sering digunakan :
 - ◇ Gradient





- ◇ Kelemahan ;
 - ▶ Untuk pixel2 terakhir tidak akan bisa didapat gradiennya
- ◇ Robert gradien



Screen clipping taken: 26/12/2011 12:04

ii. Domain frekwensi :

1) High-pass-filter

- Sharpening dilakukan dengan menyaring/ menghilangkan frekuensi rendah dari citra.

6. Image Segmentation

- a. Thresholding
- b. Region growing
- c. Split & merge

-
- Merupakan proses untuk memisahkan citra menjadi bagian-bagian pembentuknya.
 - Pendekatan algoritma :
 - Berdasarkan discontinue -> perubahan warna Mendadak : deteksi titik, garis, dan tepi
 - Berdasarkan similarity :
 - Pengelompokan berdasarkan distribusi properti pixel warna : contoh thresholding
 - Mencari region berdasarkan persamaan karakteristik suatu area : contoh region growing dan split & merge.

a. Thresholding

- Menempatkan suatu atau lebih lthreshold pada sumbu datar histogram untuk memishkan kelompok warna pixel yang diduga sebagai penyusun object.
- Jenis threshold :

$$T = T[x, y, p(x, y), f(x, y)]$$

$f(x,y)$: gray level pada titik (x,y)

$p(x,y)$: properti lokal dari titik (x,y) ; misal: gray level rata-rata dari area ketetanggaan yang berpusat di (x,y)

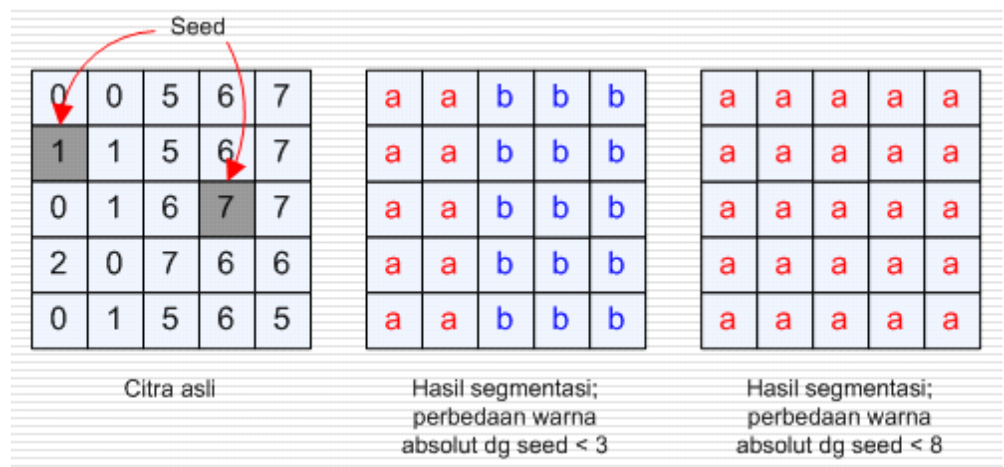
- ❑ Global → T hanya tergantung dari $f(x,y)$
- ❑ Local → T dipengaruhi oleh $f(x,y)$ dan $p(x,y)$
- ❑ Dynamic → T tergantung dari koordinat spasial titik (x,y)

Screen clipping taken: 26/12/2011 12:56

- Kelemahan :
 - ❑ Penentuan nilai threshold yang tepat
 - ❑ Bermasalah jika kemunculan warna dari citra cenderung sama, tidak dapat diprediksi batas objeknya.

b. Region growing

- Segementasi berorientasi daerah
- Segementasi membagi R menjadi R_1, R_2, \dots, R_n sampai terjadi segmentasi.
- Deskripsi region growing :
 - ❑ Prosedur yang mengelompokkan pixel atau sub-region menjadi region yang lebih besar
 - ❑ Pendekatan paling sederhana: pixel aggregation
 - Mulai dengan sekumpulan titik 'benih' (seed)
 - Dari titik-titik tsb region diperluas dengan menambahkan titik-titik tetangganya yang memiliki properti yang sama (misal: gray level, tekstur, warna)
 - Jika tidak ada lagi titik tetangga yang dapat ditambahkan lagi, maka proses untuk region tersebut dihentikan
 - ❑ Ilustrasi :



Screen clipping taken: 26/12/2011 13:03

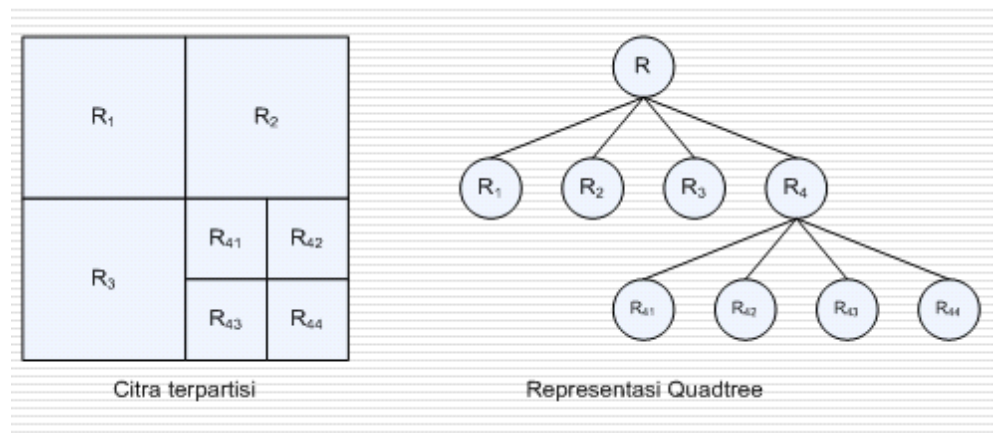
- ❑ Masalah yang ada:
 - Penentuan lokasi seed yang tepat
 - ◇ Misal : warna yang terang
 - Penentuan properti yang tepat untuk mengelompokkan titik menjadi

region

- ◇ Misal : tekstur
- Kondisi penghenti
 - ◇ Misal : jika tidak ada lagi titik tetangga yang memenuhi syarat.

c. Split & merge

- Membagi citra menjadi sekumpulan region acak yang disjoint kemudian menggabungkan atau kembali membaginya hingga terpenuhi syarat segmentasi (a) – (e)
- Algoritma bersifat rekursif
- Memanfaatkan quadtree
- Ilustrasi :



Screen clipping taken: 26/12/2011 13:10

d. Contoh aplikasi Image segmentation :

- i. Magic Wand

7. Kompresi Citra

- Materi :
 - i. Klasifikasi Kompresi
 - ii. Teknik Kompresi
- Penjabaran :
 - Teknik kompresi yang diharapkan :
 - Proses kompresi/dekompresi yang cepat
 - Membutuhkan memori yang kecil
 - Kualitas yang bagus
 - Proses tranfer dan penyimpanan yang mudah
 - Teknik kompresi :
 - Berdasarkan hasil dapat dibagi 2 :
 - Lossles compression
 - Lossy compression
 - Berdasarkan teknik :
 - Entropy Encoding (Lossless)
 - ◇ Run Length Encoding (RLE)
 - ◇ Pattern substitusi
 - ◇ Huffman
 - ◇ DPCM
 - Source Encoding (lossy)
 - ◇ Quantizing Compression
 - ◇ Transform Encoding

- Hybrid Encoding (lossy)
 - ◇ JPEG

- Penjelasan Teknik kompresi :

- Run Length Encoding (RLE)

- Mengubah matrik nilai menjadi urutan nilai dimana nantinya jumlah kemunculan dari suatu nilai di hitung dan digantikan dengan indeks (nilai, jumlah)
 - Misal :

1	2	1	1	1	1
1	3	4	4	4	4
1	1	3	3	3	5
1	1	1	1	3	3

Diubah dalam bentuk sekuensial
 → 1 2 1 1 1 1 3 4 4 4 4 1 1 3 3 3 5 1 1 1 1 3 3 = 24 byte

Dihitung jumlah kemunculan data
 → (1,1) (2,1) (1,5) (3,1) (4,4) (1,2) (3,3) (5,1) (1,4) (3,2)

Data Kompresi
 → 1 1 2 1 1 5 3 1 4 4 1 2 3 3 5 1 1 4 3 2 = 20 byte

Screen clipping taken: 26/12/2011 13:29

- Masalah :

- ◇ Data bisa menjadi mengecil atau malah menjadi membesar.

- Shannon's Source Coding Theory

- Mengubah nilai simbol menjadi sebuah simbol lainnya (misal biner)
 - Semakin besar kemunculan suatu nilai, maka akan dikodekan dengan biner yang paling kecil.
 - Rumus probabilitas entropy :

$$\diamond I(a_i) = \log_2 \frac{1}{p(a_i)} = -\log_2 p(a_i).$$

- Entropy seluruh data:

$$\diamond E = \sum_{i=1}^N p(a_i) I(a_i) = - \sum_{i=1}^N p(a_i) \log_2 p(a_i).$$

- Huffman Code

- Memberikan kode biner berdasar huffman tree yang dibuat, dimana data yang paling sering muncul diberi kode yang paling kecil.
 - Contoh :

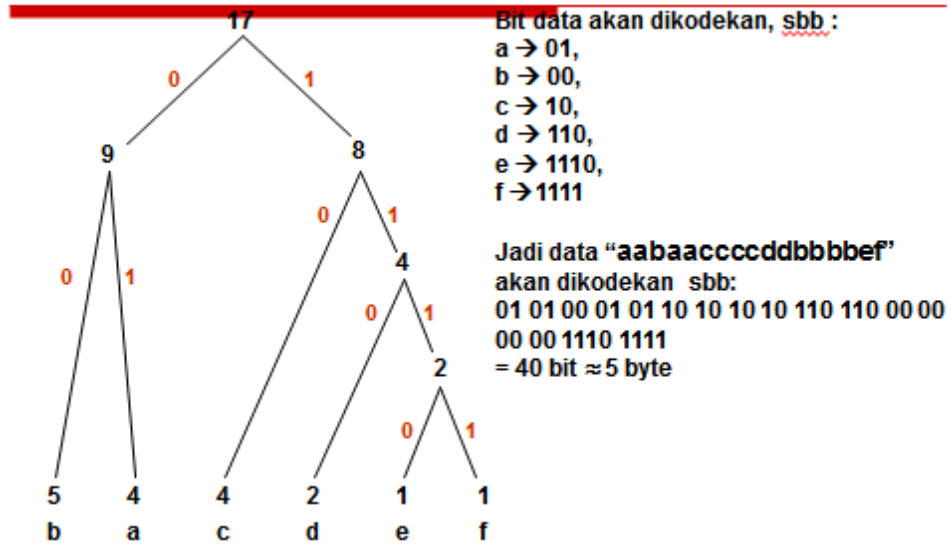
- Misalkan : aabaacccddbbbbbef = 17 byte

- Pertama-tama hitung frekuensi masing-masing data :

- a → 4, b → 5, c → 4, d → 2, e → 1, f → 1

- Masing-masing data diberi kode → bit data yang sering muncul diberi kode lebih kecil dibandingkan dengan yang jarang muncul.

- b → 5, a → 4, c → 4, d → 2, e → 1, f → 1



- Quantizing Compresiion
 - Merupakan cara kompresi dengan mengubah isi data dengan nilai baru, diman data hasil compresi tidak dapat dikembalikan lagi.
 - Misal :

2	9	6	4	8	2	6	3	8	5	9	3	7
3	8	5	4	7	6	3	8	2	8	4	7	3
3	8	4	7	4	9	2	3	8	2	7	4	9
3	9	4	7	2	7	6	2	1	6	5	3	0
2	0	4	3	8	9	5	4	7	1	2	8	3

• Histogram :

- Warna 0 = 2
- Warna 1 = 2
- Warna 2 = 9
- Warna 3 = 11
- Warna 4 = 9
- Warna 5 = 4
- Warna 6 = 5
- Warna 7 = 8
- Warna 8 = 9
- Warna 9 = 6

} Dikodekan menjadi 0 (Jumlahnya 13 pixel)

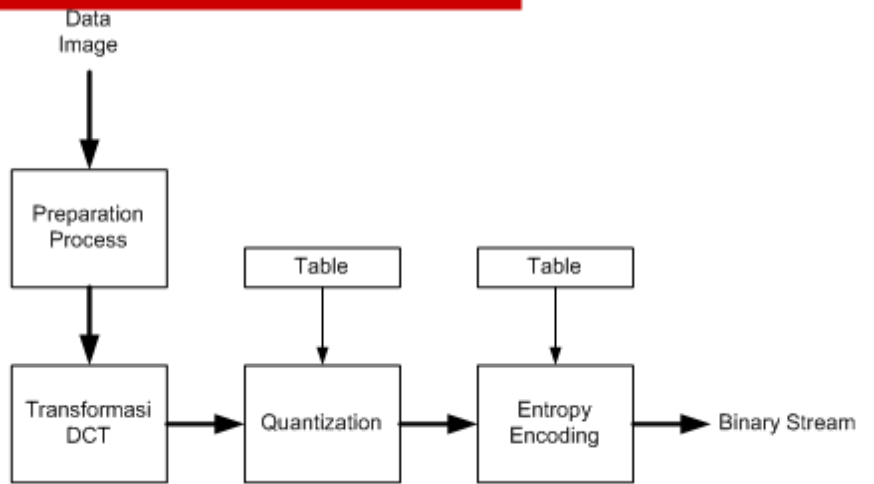
} Dikodekan menjadi 1 (Jumlahnya 20 pixel)

} Dikodekan menjadi 2 (Jumlahnya 17 pixel)

} Dikodekan menjadi 3 (Jumlahnya 15 pixel)

Screen clipping taken: 26/12/2011 13:26

- JPEG (Joint Photographic Experts Group)
 - Tabel proses :



Screen clipping taken: 26/12/2011 13:43

- Tahapan dalam JPEG

◇ Tahap Persiapan :

- ▶ Dalam proses ini dilakukan proses membagi citra menjadi blok 8x8

◇ Transformasi DCT

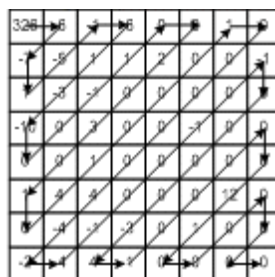
- ▶ Mengubah/ menghitung frekuensi pembentuk dari citra blok 8x8 dan memisahkan frekuensi rendah dan frekuensi tinggi dari hasil transformasi DCT

◇ Quantisasi (lossy)

- ▶ Proses ini bertujuan untuk menghilangkan nilai yang tidak penting (frekuensi tinggi) pada matrik hasil dari transformasi DCT.

◇ Entropy Encoding (lossless)

- ▶ Entropy Encoding adalah teknik kompresi yang bersifat lossless. Tahap ini bertujuan untuk mengkompresi matrik hasil quantisasi, bisa menggunakan metode huffman atau RLE
- ▶ Proses Entropy Encoding terhadap hasil quantisasi di atas dengan pembacaan zig-zag :



Hasil encoding jika menggunakan RLE :
 326,-6,-7,1,-5,1,6,1,-3, [0,3] , -1,1,[0,2],2,[0,1],3,
 [0,1], 1, [0,1],4,1,[0,3],1,[0,5],4,-4,-2,4,-1,[0,2],-
 1,[0,1], -1, [0,4],-3,4,1,[0,5],12,1,[0,7] = 49 byte

8. Steganography

- Steganography adalah Teknik menyembunyikan data rahasia di dalam media digital sehingga keberadaannya tidak diketahui orang.
- Steganography membutuhkan 2 properti :
 - Media Penampung
 - Data yang akan disembunyikan
- Kegunaan Steganography :
 - Data Hiding, bertujuan untuk menyamarkan eksistensi (keberadaan) data rahasia

- sehingga sulit dideteksi.
- Watermarking, bertujuan melindungi hak cipta suatu produk
- Kriteria Steganography
 - Media ataupun data yang disembunyikan bisa berupa teks,citra,audio maupun video.
 - Kriteria yang harus diperhatikan dalam penyembunyian data :
 - Fidelity, Mutu citra penampung tidak jauh berubah
 - Setelah penambahan data rahasia, citra hasil steganography masih terlihat baik. Pengamat tidak mengetahui kalau di dalam citra tersebut terdapat data rahasia
 - Robustness
 - Data yang disembunyikan harus tahan (robust) terhadap berbagai operasi manipulasi yang dilakukan pada citra penampung, seperti perubahan kontras,penajaman,kompresi,rotasi,zooming,croping,enkripsi, dsb.
 - Recovery
 - Data yang disembunyikan harus dapat diekstraksi kembali.

- Data Hidding

- Teknik penyembunyian data
 - Penyembunyian data dilakukan dengan mengganti bit-bit data di dalam segmen citra dengan bit-bit data rahasia.
 - Metode yang paling sederhana adalah metode adalah modifikasi LSB (Least Significant Bit Modification)
 - Pada susunan bit di dalam sebuah byte (1 byte = 8 bit), terdapat :
 - ◆ MSB (Most Significant Bit), bit yang paling berarti
 - ◆ LSB (Least Significant Bit), bit yang kurang berarti
 - Contoh :
 - ◆ *1101100**0
 - *= MSB
 - **= LSB
 - Contoh :

■ Data Penampung :

□ Dalam byte = {10,10,14,21}

□ Dalam bit = {00001010,00001010,00001110,00010101}

■ Data rahasia : 14 (dalam bit : 1110)

■ Hasil Steganography :

□ Dalam bit = {00001011,00001011,00001111,00010100}

□ Dalam byte = {11, 11, 15, 20}

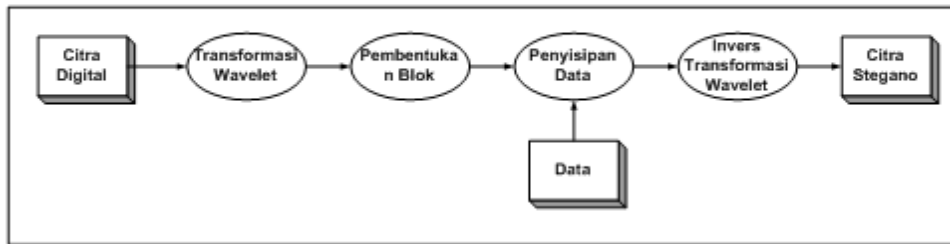
Screen clipping taken: 26/12/2011 14:32

- Watermarking

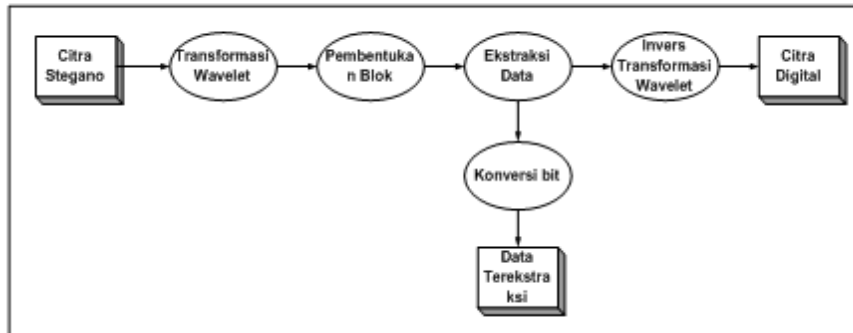
- Watermarking merupakan salah satu bentuk aplikasi dari steganography
- Perbedaannya :
 - Pada Data Hiding, informasi rahasia disembunyikan di dalam media digital dimana media penampungnya tidak berarti apa-apa.
 - Pada Watermarking justru media digital tersebut yang akan dilindungi kepemilikannya dengan pemberian label hak cipta (watermark).

- Steganography dengan Transformasi Wavelet

Penyisipan Data



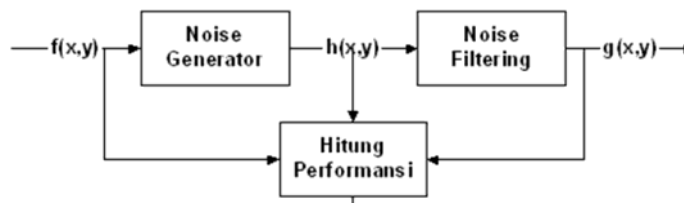
Ekstraksi Data



- Ada beberapa hal yang dapat menjadi fokus penelitian dalam steganography
 - Teknik penyisipan
 - Lokasi penyisipan
 - Peningkatan kriteria

9. Fidelity Criteria

- Sulit dilakukan pengukuran secara objektif terhadap beberapa metode pengolahan citra.
- Diperlukan suatu mekanisme dimana kita dapat mengukur performansi dari suatu proses pengolahan citra.
- Pengukuran tergantung pada proses dan tujuan pengolahan
 - Misalkan:
 - Noise Filtering => menghilangkan noise
 - Image Compression => menurunkan ukuran file dan menjaga kualitas
 - Deteksi sisi => menghasilkan sisi
- Untuk menghitung ferformansi dapat dilakukan skenario seperti :



- Parameter pengukuran objektif yang sering digunakan dalam proses noise filtering adalah pengukuran MSE dan SNR

□ Misalkan

- $f(x,y)$ = data citra awal
- $h(x,y)$ = data citra ter-noise
- $g(x,y)$ = data citra hasil filter

□ MSE untuk $h(x,y)$:

$$MSE = \frac{\sum_{x=1}^M \sum_{y=1}^N e^2[x,y]}{M \cdot N} = \frac{\sum_{x=1}^M \sum_{y=1}^N [h[x,y] - f[x,y]]^2}{M \cdot N}$$

Screen clipping taken: 26/12/2011 14:53

□ SNR dari $h(x,y)$ dapat dihitung:

$$SNR = 10 \log_{10} \left[\frac{\sum_{x,y} f^2[x,y]}{\sum_{x,y} [f[x,y] - h[x,y]]^2} \right]$$

□ PSNR dari $h(x,y)$ dapat dihitung:

$$PSNR = 20 \log_{10} \frac{255}{MSE^{1/2}}$$

Screen clipping taken: 26/12/2011 14:53

- Selain pengukuran objektif juga ada pengukuran subjektif
 - Pengukuran subjektif :kriteria ditentukan berdasarkan pengamatan mata manusia, sehingga kualitas subyektif tergantung kepada persepsi visual pengamat
 - Agar lebih valid maka untuk pengukuran subjektif biasanya dilakukan oleh seorang pakar (tergantung kasus)
- Fidelity Criteria untuk Image Compression
- Terdapat 2 tujuan utama yaitu: mengecilkan ukuran file tetapi tetap menjaga kualitas gambar
 - Dimana 2 tujuan ini pada umumnya saling bertolak belakang (trade-off)
 - Ukuran file kecil, kualitas jelek
 - Ukuran besar, kualitas bagus
 - Problem: menentukan posisi treshold dari 2 komponen trade-off

- Untuk masalah ukuran dilakukan perhitungan Ratio Kompresi:

$$RK = \left[1 - \frac{D'}{D} \right] \times 100$$

- Dimana:

- D' = ukuran data hasil kompresi
- D = ukuran data sebelum kompresi

Screen clipping taken: 26/12/2011 15:02

- Untuk kualitas gambar perhitungan hanya dilakukan untuk kompresi yang bersifat lossy, dengan :
 - Menghitung MSE dan PSNR
 - Menghitung BER (bit error rate) bit yang tidak sesuai dengan data asli.
- Untuk kompresi yang bersifat lossless dapat dilakukan perhitungan waktu eksekusi dan penentuan kompleksitas metode yang digunakan
- Pada umumnya untuk menghasilkan ukuran file yang kecil metode Compressi menggunakan file pemetaan yang sering disebut dengan codebook

- Fidelity Criteria untuk Deteksi Sisi

- Salah satu pengukuran yang dapat dilakukan adalah

$$Akur. = \frac{B - S}{Tot_{Sisi_{ori}}} \times 100$$

- Dimana:

- B = jumlah pixel sisi yang benar
- S = jumlah pixel sisi yang salah (harusnya sisi dikatakan tidak atau harusnya tidak sisi dikatakan sisi)

- Pendekatan lainnya dapat dilakukan dengan menganggap Proses Deteksi Sisi sebagai masalah klasifikasi
- Kita dapat menggunakan parameter pengukuran proses klasifikasi dalam data mining seperti Precision dan Recall

Good-luck