

# KOMPRESI CITRA

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.

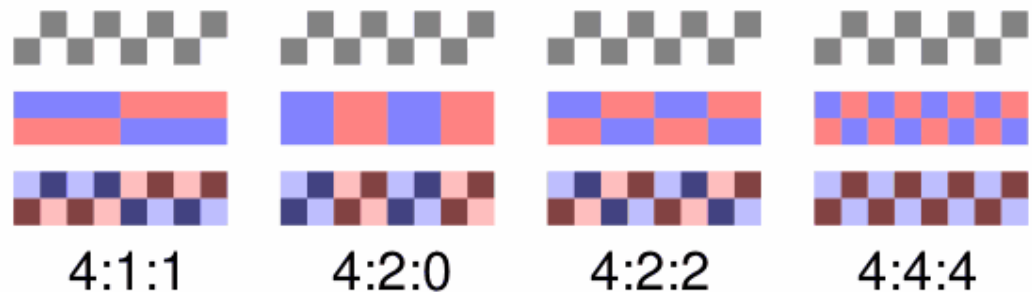
## TEKNIK KOMPRESI CITRA

Teknik kompresi pada citra tetap sama:

- **Lossy Compression:**

- o Ukuran file citra menjadi lebih kecil dengan menghilangkan beberapa informasi dalam citra asli.
- o Teknik ini mengubah detail dan warna pada file citra menjadi lebih sederhana tanpa terlihat perbedaan yang mencolok dalam pandangan manusia, sehingga ukurannya menjadi lebih kecil.
- o Biasanya digunakan pada citra foto atau image lain yang tidak terlalu memerlukan detail citra, dimana kehilangan bit rate foto tidak berpengaruh pada citra.
- o Beberapa teknik loseless:
  - **Color reduction:** untuk warna-warna tertentu yang mayoritas dimana informasi warna disimpan dalam color palette.

- **Chroma subsampling:** teknik yang memanfaatkan fakta bahwa mata manusia merasa brightness (luminance) lebih berpengaruh daripada warna (chrominance) itu sendiri, maka dilakukan pengurangan resolusi warna dengan disampling ulang. Biasanya digunakan pada sinyal YUV.
  - Chroma Subsampling terdiri dari 3 komponen: Y (luminance) : U (CBlue) : V (CRed)



- **Transform coding:** menggunakan Fourier Transform seperti DCT.
  - Fractal Compression: adalah suatu metode lossy untuk mengompresi citra dengan menggunakan kurva fractal. Sangat cocok untuk citra natural seperti pepohonan, pakis, pegunungan, dan awan.
  - Fractal Compression bersandar pada fakta bahwa dalam sebuah image, terdapat bagian-bagian image yang menyerupai bagian bagian image yang lain.
  - Proses kompresi Fractal lebih lambat daripada JPEG sedangkan proses dekompresinya sama.
- **Lossless Compression:**
  - Teknik kompresi citra dimana tidak ada satupun informasi citra yang dihilangkan.
  - Biasa digunakan pada citra medis.
  - Metode loseless: Run Length Encoding, Entropy Encoding (Huffman, Aritmatik), dan Adaptive Dictionary Based (LZW)

## HAL-HAL PENTING DALAM KOMPRESI CITRA

### 1. Scalability/Progressive Coding/Embedded Bitstream

- Adalah kualitas dari hasil proses pengkompresian citra karena manipulasi bitstream tanpa adanya dekompresi atau rekompresi.
  - Biasanya dikenal pada loseless codec.
  - Contohnya pada saat preview image sementara image tersebut didownload. Semakin baik scalability, makin bagus preview image.
  - Tipe scalability:
    - a. **Quality progressive:** dimana image dikompres secara perlahan-lahan dengan penurunan kualitasnya
    - b. **Resolution progressive:** dimana image dikompresi dengan mengkode resolusi image yang lebih rendah terlebih dahulu baru kemudian ke resolusi yang lebih tinggi.
    - c. **Component progressive:** dimana image dikompresi berdasarkan komponennya, pertama mengkode komponen gray baru kemudian komponen warnanya.
2. **Region of Interest Coding:** daerah-daerah tertentu diencode dengan kualitas yang lebih tinggi daripada yang lain.
3. **Meta Information:** image yang dikompres juga dapat memiliki meta information seperti statistik warna, tekstur, small preview image, dan author atau copyright information

## PENGUKURAN ERROR KOMPRESI CITRA

Dalam kompresi image terdapat suatu standar pengukuran error (galat) kompresi:

- MSE (**Mean Square Error**), yaitu sigma dari jumlah error antara citra hasil kompresi dan citra asli.

$$MSE = \frac{1}{MN} \sum_{y=1}^M \sum_{x=1}^N [I(x,y) - I'(x,y)]^2$$

Dimana:  $I(x,y)$  adalah nilai pixel di citra asli

$I'(x,y)$  adalah nilai pixel pada citra hasil kompresi

M,N adalah dimensi image

- **Peak Signal to Noise Ratio (PSNR)**, yaitu untuk menghitung peak error.

$$PSNR = 20 * \log_{10} (255 / \sqrt{MSE})$$

Nilai MSE yang rendah akan lebih baik, sedangkan nilai PSNR yang tinggi akan lebih baik.

## ALGORITMA KOMPRESI/DEKOMPRESI CITRA

Algoritma umum untuk kompresi image adalah:

1. Menentukan bitrate dan toleransi distorsi image dari inputan user.
2. Pembagian data image ke dalam bagian-bagian tertentu sesuai dengan tingkat kepentingan yang ada (**classifying**).

Menggunakan salah satu teknik: DWT (Discrete Wavelet Transform) yang akan mencari frekuensi nilai pixel masing-masing, menggabungkannya menjadi satu dan mengelompokkannya sebagai berikut:

|           |           |           |           |           |           |           |           |           |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| <b>LL</b> | <b>HL</b> | <b>LL</b> | <b>HL</b> | <b>HL</b> | <b>LL</b> | <b>HL</b> | <b>HL</b> | <b>HL</b> |
|           |           | <b>LH</b> | <b>HH</b> |           | <b>LH</b> | <b>HH</b> |           |           |
| <b>LH</b> | <b>HH</b> | <b>LH</b> |           | <b>HH</b> | <b>LH</b> |           | <b>HH</b> |           |

(a) Single Level Decomposition      (b) Two Level Decomposition      (c) Three Level Decomposition

Dimana    LL : Low Low Frequency (most importance)  
            HL : High Low Frequency (lesser importance)  
            LH : Low High Frequency (more lesser importance)  
            HH : High High Frequency (most less importance)



Hasil dekomposisi 3 level decomposition

3. Pembagian bit-bit di dalam masing-masing bagian yang ada (**bit allocation**).
4. Lakukan kuantisasi (**quantization**).
  - o Kuantisasi Scalar : data-data dikuantisasi sendiri-sendiri
  - o Kuantisasi Vector : data-data dikuantisasi sebagai suatu himpunan nilai-nilai vektor yang diperlakukan sebagai suatu kesatuan.
5. Lakukan pengenkodingan untuk masing-masing bagian yang sudah dikuantisasi tadi dengan menggunakan teknik entropy coding (huffman dan aritmatik) dan menuliskannya ke dalam file hasil.

Sedangkan algoritma umum dekompresi image adalah:

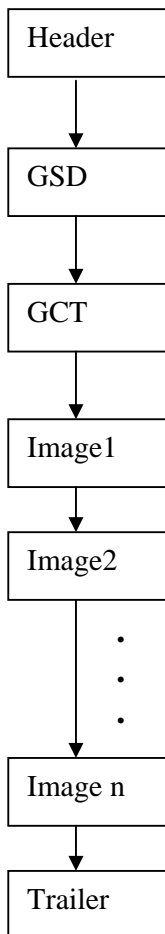
1. Baca data hasil kompresi menggunakan entropy dekoder.
2. Dekuantisasi data.
3. Rebuild image.

### **BEBERAPA METODE KOMPRESI CITRA**

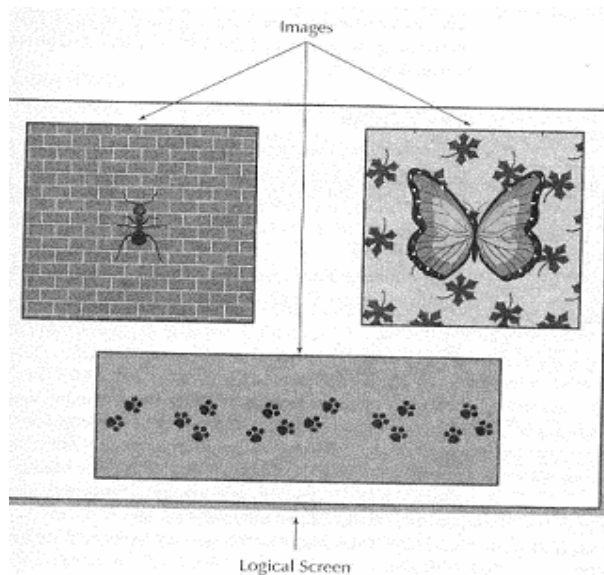
| Algoritma | BMP | GIF | PNG | JPEG |
|-----------|-----|-----|-----|------|
| RLE       | X   |     |     | X    |
| LZ        |     | X   | X   |      |
| Huffman   |     |     | X   | X    |
| DCT       |     |     |     | X    |

## TEKNIK KOMPRESI GIF

- GIF (Graphic Interchange Format) dibuat oleh CompuServe pada tahun 1987 untuk menyimpan berbagai file bitmap menjadi file lain yang mudah diubah dan ditransmisikan pada jaringan komputer.
- GIF merupakan format citra web yang tertua yang mendukung kedalaman warna sampai 8 bit (256 warna), menggunakan 4 langkah interlacing, mendukung transparency, dan mampu menyimpan banyak image dalam 1 file.
- Byte ordering: LSB – MSB
- Struktur file GIF:



- o Header: menyimpan informasi identitas file GIF (3 bytes, harus string "GIF") dan versinya (3 bytes, harus string "87a" or "89b")
- o Global Screen Descriptor: mendefinisikan logical screen area dimana masing-masing file GIF ditampilkan.



- o Global Color Table: masing-masing image dalam GIF dapat menggunakan global color table atau tabel warnanya sendiri-sendiri. Penggunaan GCT akan memperkecil ukuran file GIF.
- o Image1, Image2, Image3, ... Image-n: dimana masing-masing image memiliki struktur blok sendiri-sendiri dan terminator antar file.

- o Trailer: Akhir dari sebuah file GIF
- Kompresi GIF menggunakan teknik LZW: gambar GIF yang berpola horizontal dan memiliki perubahan warna yang sedikit, serta tidak bernoise akan menghasilkan hasil kompresan yang baik.

```

Global String DICTIONARY [0..511]
Global NEXTCODE = 256

Procedure Initialize
Begin
  For I = 0 To NEXTCODE - 1 Do
    DICTIONARY [I] = CHARACTER (I)
  End
End

Function SearchDictionary (String SEARCH)
Begin
  For I = 0 To NEXTCODE - 1 Do
    Begin
      IF DICTIONARY [I] = SEARCH Then
        Return I
      End
    End
  Return -1
End

Procedure Compress (String DATA)
Begin
  Initialize
  LASTSTRING = NULL
  For I = 1 To Length (DATA) Do
    Begin
      CurrentString = LASTSTRING + DATA [I]
      CODE = SearchDictionary (CURRENTSTRING)
      IF CODE < 0 Then
        Begin
          // We now have a string with no match in the
          // dictionary. Output the code for the longest
          // string with a match.
          CODE = SearchDictionary (LASTSTRING)
          Output (CODE)
          // Add the nonmatching string to the dictionary.
          DICTIONARY [NEXTCODE] = CURRENTSTRING
          NEXTCODE = NEXTCODE + 1
          // Start a new string, beginning at the point
          // where we no longer had a match.
          LASTSTRING = DATA [I]
        End
      Else
        Begin
          // The current string has a match in the dictionary.
          // Keep adding to it until there is no match.
          LASTSTRING = CURRENTSTRING
        End
      End
    End
  // Output what is left over.
  Output (SearchDictionary (LASTSTRING))
End

```

## A MAN A PLAN A CANAL PANAMA

| Input                       | Output  | New Code              |
|-----------------------------|---------|-----------------------|
| A MAN A PLAN A CANAL PANAMA | A       |                       |
| MAN A PLAN A CANAL PANAMA   | <SPACE> | 256='A<SPACE>'        |
| MAN A PLAN A CANAL PANAMA   | M       | 257='<SPACE>M'        |
| AN A PLAN A CANAL PANAMA    | A       | 258='MA'              |
| N A PLAN A CANAL PANAMA     | N       | 259='AN'              |
| A PLAN A CANAL PANAMA       | <SPACE> | 260='N<SPACE>'        |
| A PLAN A CANAL PANAMA       | 256     | 261='A'               |
| PLAN A CANAL PANAMA         | P       | 262='A<SPACE>P'       |
| LAN A CANAL PANAMA          | L       | 263='PL'              |
| AN A CANAL PANAMA           | 259     | 264='LA'              |
| A CANAL PANAMA              | 261     | 265='AN<SPACE>'       |
| CANAL PANAMA                | <SPACE> | 266='<SPACE>A<SPACE>' |
| CANAL PANAMA                | C       | 267='<SPACE>C'        |
| ANAL PANAMA                 | 259     | 268='CA'              |
| AL PANAMA                   | A       | 269='ANA'             |
| L PANAMA                    | L       | 270='AL'              |
| PANAMA                      | <SPACE> | 271='L<SPACE>'        |
| PANAMA                      | P       | 272='<SPACE>P'        |
| ANAMA                       | 269     | 273='PA'              |
| MA                          | 258     | 274='ANAM'            |

- LZW kurang baik digunakan dalam bilevel (hitam-putih) dan true color
- Format file GIF:
  - o GIF87a: mendukung interlacing dan mampu menyimpan beberapa image dalam 1 file, ditemukan tahun 1987 dan menjadi standar.
  - o GIF89a: kelanjutan dari 87a dan ditambahkan dengan dukungan transparency, mendukung text, dan animasi.
- Animated GIF: tidak ada standar bagaimana harus ditampilkan sehingga umumnya image viewer hanya akan menampilkan image pertama dari file GIF. Animated GIF memiliki informasi berapa kali harus diloop.
- Tidak semua bagian dalam animated GIF ditampilkan kembali, hanya bagian yang berubah saja yang ditampilkan kembali.

## TEKNIK KOMPRESI PNG

- PNG (Portable Network Graphics) digunakan di Internet dan merupakan format terbaru setelah GIF, bahkan menggantikan GIF



untuk Internet image karena GIF terkena patent LZW yang dilakukan oleh Unisys.

- Menggunakan teknik loseless dan mendukung:
  - o Kedalaman warna 48 bit
  - o Tingkat ketelitian sampling: 1,2,4,8, dan 16 bit
  - o Memiliki alpha channel untuk mengontrol transparency
  - o Teknik pencocokan warna yang lebih canggih dan akurat
- Diprakarsai oleh Thomas Boutell dari PNG Development Group, dan versi finalnya direlease pada 1 Oktober 1996, 1,5 tahun sejak project berjalan.
- Byte ordering: MSB-LSB
- Format penamaan file PNG diatur ke dalam suatu urutan blok biner yang disebut sebagai "chunk" (gumpalan), yang terdiri dari:
  - o Length (4 bytes), berupa informasi ukuran PNG
  - o Type (4 byte), berupa informasi nama chunk

Nama chunk terdiri dari 4 karakter ASCII dengan spesifikasi:

- Karakter ke-1,2, dan 4 boleh uppercase/lowercase
- Jika karakter ke-1 uppercase, berarti **critical chunk** (harus valid), contohnya: IHDR, PLTE, IDAT, dan IEND.
- Jika karakter ke-1 lowercase, berarti **non-critical chunk** (contohnya: bKGD, cHRM, gAMA, hIST, pHYS, sBIT, tEXt, tIME, tRNS, zTXt)
- Jika karakter ke-2 uppercase, berarti public (PNG Standard)
- Jika karakter ke-2 lowercase, berarti private PNG
- Jika karakter ke-4 lowercase, berarti save-to-copy
- Jika karakter ke-4 uppercase, berarti unsave-to-copy
- Karakter 3 harus uppercase

Contoh penamaan:

IHDR: critical, public, unsafe to copy

gAMA: noncritical, public, unsafe to copy

pHYs: noncritical, public, safe to copy

apPx: noncritical, private, safe to copy

A1PX: invalid

ApPx: critical, private, safe to copy

apPX: noncritical, private, unsafe to copy

aaaX: invalid

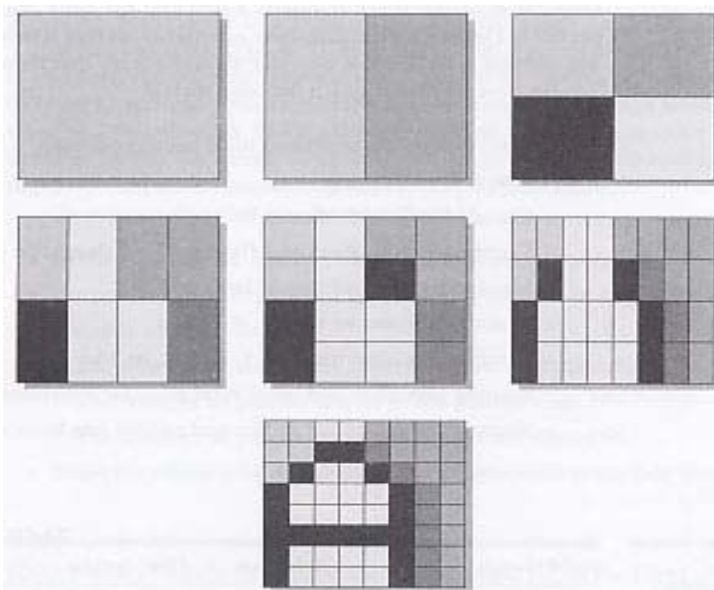
- Data (ukuran dinamis), berupa data PNG.
- CRC (Cyclic Redundancy Check), berupa CRC-32 untuk pendeteksian error checking pada saat transmisi data.

Proses PNG decoder adalah sebagai berikut:

- Baca chunk data size
- Baca dan simpan chunk type
- Jika ukuran chunk data lebih besar daripada data buffer, alokasikan buffer yang lebih besar
- Baca chunk data
- Hitung CRC value dari chunk data
- Baca CRC dari file yang diterima
- Bandingkan hasil perhitungan CRC dengan CRC dari file, jika tidak sama, berarti chunk invalid, minta kirim ulang.

- Sedangkan struktur file PNG adalah:
  - PNG Signature: tanda file PNG
  - IHDR chunk: menyimpan dimension, depth, dan color type
  - PLTE chunk: untuk PNG yang menggunakan color palette type
  - IDAT chunk 1, IDAT chunk 2, IDAT chunk 3, ... IDAT chunk-n
  - IEND chunk: end of PNG image
- PNG mendukung 5 cara untuk merepresentasikan warna, dimana tipe warna disimpan dalam bagian IHDR chunk:
  - RGB Triple (R,G, dan B): untuk 8 atau 16 bits
  - Color Palette: yang disimpan dalam PLTE chunk dengan bit depth 1,2,4 atau 8.

- Grayscale: 1 komponen warna per image, bisa digunakan untuk semua bit depth.
- RGB Alpha Channel:
  - agar image dan background dapat dikombinasikan
  - untuk mengontrol transparency
  - hanya bisa digunakan pada bit depth 8 atau 16 bits
  - jika alpha channel 0 berarti 100% transparan, sehingga background terlihat seutuhnya.
  - Jika alpha channel  $2^{\text{image bit depth}} - 1$  berarti fully opaque, sehingga background sama sekali tidak terlihat karena tertutup oleh image.
- Grayscale with Alpha Channel: hanya bisa 8/16 bits
- PNG mendukung interlacing yang disebut Adam 7, yang menginterlace berdasarkan pixel daripada berdasarkan baris. Adam akan membagi image ke dalam 8x8 pixel, yang akan diupdate dalam 7 fase interlacing sebagai berikut:



- Teknik kompresi yang digunakan adalah Deflate yang merupakan kelanjutan dari algoritma Lempel-Ziv. Cara kerja Deflate sama dengan LZW dan melakukan scanning secara horisontal.

## TEKNIK KOMPRESI JPEG

- JPEG (Joint Photographic Experts Group) menggunakan teknik kompresi lossy sehingga sulit untuk proses pengeditan.
- JPEG cocok untuk citra pemandangan (natural generated image), tidak cocok untuk citra yang mengandung banyak garis, ketajaman warna, dan computer generated image
- JPEG's compression models:

- o **Sequential:** kompresi dilakukan secara top-down, left-right menggunakan proses single-scan dan algoritma Huffman Encoding 8 bit secara sekuensial

```
While MORESCANS do
Begin
    ReadScanData
End
PerformDCT
ColorConvert
DisplayImage
```

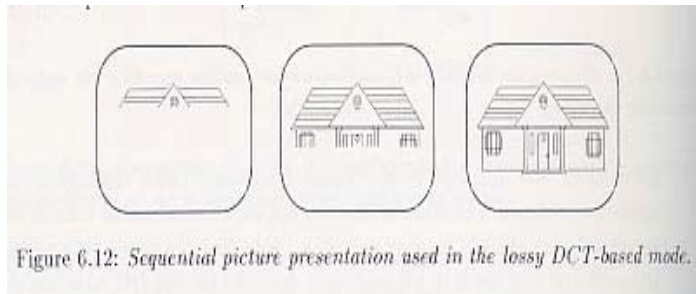
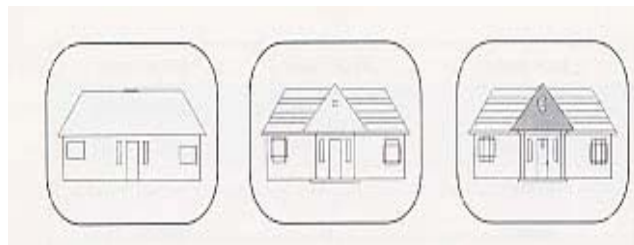


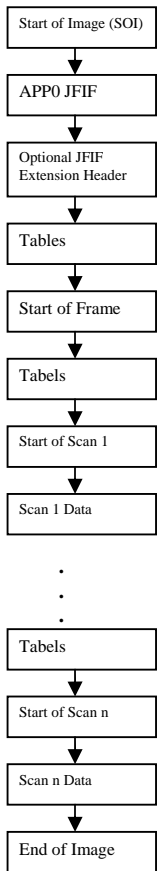
Figure 6.12: Sequential picture presentation used in the lossy DCT-based mode.

- o **Progressive:** kompresi dilakukan dengan multiple-scan secara progresif, sehingga kita dapat mengira-ira gambar yang akan kita download.

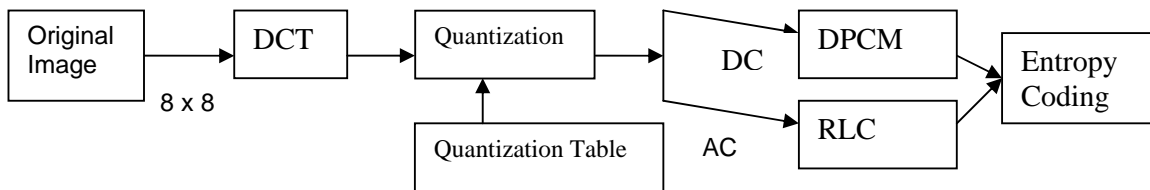
```
While MORESCANS do
Begin
    ReadScanData
    PerformDCT
    ColorConvert
    DisplayImage
End
```



- o **Hierarchical:** super-progressive mode, dimana image akan dipecah-pecah menjadi sub image yang disebut frame. Frame pertama akan membentuk image dalam resolusi rendah hingga berangsur-angsur ke resolusi tinggi.
- o Loseless (JPEG-LS): exact image
- JPEG merupakan nama teknik kompresi, sedangkan nama format filenya adalah JFIF (JPEG File Interchange Format)



- Tingkat kompresi yang baik untuk JPEG adalah 10:1-20:1 untuk citra foto, 30:1-50:1 untuk citra web, dan 60:1-100:1 untuk kualitas rendah seperti citra untuk ponsel.
- Byte order: MSB-LSB
- Tahapan kompresi JPEG:
  - o Sampling: adalah proses pengkonversian data pixel dari RGB ke YUV/YIQ dan dilakukan down sampling. Biasanya sampling dilakukan per 8x8 blok, semakin banyak blok yang dipakai makin bagus kualitas sampling yang dihasilkan.
  - o DCT (Discreate Cosine Transform) : hasil dari proses sampling akan digunakan sebagai inputan proses DCT, dimana blok 8x8 pixels akan diubah menjadi fungsi matriks cosinus
  - o Quantization: proses membersihkan koefisien DCT yang tidak penting untuk pembentukan image baru. Hal ini yang menyebabkan JPEG bersifat lossy.
  - o Entropy Coding: proses penggunaan algoritma entropy, misalnya Huffman atau Aritmatik untuk mengkodekan koefisien hasil proses DCT yang akan mengeliminasi nilai-nilai matriks yang bernilai nol secara zig-zag order.



- Dalam JPEG terdapat beberapa “marker” sebagai tanda yang memisahkan antar komponennya yang berukuran 2 bytes, dimana byte pertama selalu bernilai FF<sub>16</sub> sedangkan bit kedua bisa berupa:
  - o APP<sub>n</sub>: untuk handle application specific data, misalnya informasi tambahan yang ada dalam JPEG

- COM (Comment): untuk memberikan komentar plain text string seperti copyright.
- DHT (Define Huffman Table): menyimpan tabel kode-kode Algoritma Huffman
- DRI (Define Restart Interval): sebagai tanda restart interval
- DQT (Define Quantization Table): mendefinisikan tabel kuantisasi yang digunakan dalam proses kompresi
- EOI (End of Image): tanda akhir file JPEG
- RST<sub>n</sub>: restart marker
- SOI (Start of Image): tanda awal image
- SOF<sub>n</sub>: start of frame
- SOS: start of scan
- Secara umum JPEG/JFIF file menyimpan informasi:
  - Signature untuk mengidentifikasi JPEG file
  - Colorspace
  - Pixel density
  - Thumbnails
  - Relationship of pixels to sampling frequency

## **JPEG 2000**

- Adalah pengembangan kompresi JPEG.
- Didesain untuk internet, scanning, foto digital, remote sensing, medical imegrey, perpustakaan digital dan e-commerce
- Kelebihan:
  - Dapat digunakan pada bit-rate rendah sehingga dapat digunakan untuk network image dan remote sensing
  - Menggunakan Lossy dan loseless tergantung kebutuhan bandwidth. Loseless digunakan untuk medical image
  - Transmisi progresif dan akurasi & resolusi pixel tinggi
  - Menggunakan Region of Interest (ROI)

- Robustness to bit error yang digunakan untuk komunikasi jaringan dan wireless
- Open architecture: single compression/decompression
- Mendukung protective image security: watermarking, labeling, stamping, dan encryption
- Mendukung image ukuran besar 64k x 64k, size up to  $2^{32} - 1$
- Mendukung meta data dan baik untuk computer-generated imagenary. Dulu JPEG standar baik untuk natural imagenary.