

## Standard Kompresi Citra: JPEG

### Kompresi/Coding Citra (JPEG)

- Dirancang oleh Joint Photographic Experts Group (usaha kolaboratif ITU-T dan ISO)
- Mendukung macam aplikasi kompresi
  - paling umum digunakan untuk gambar diam
- Dapat mengkodekan full color image
  - rasio kompresi rata-rata 15:1
- Standar sangat fleksibel (parametrisable)
  - trade-off antara kompresi dan kualitas dapat ditentukan sesuai aplikasi

## Kompresi/Coding Citra (JPEG)

- Contoh aplikasi yang didukung algoritma JPEG:
  - Adobe's Postscript
  - ISO Office Document Architecture & Interchange format
  - ITU-T colour facsimile
  - Pertukaran citra, WWW, photography digital

## JPEG

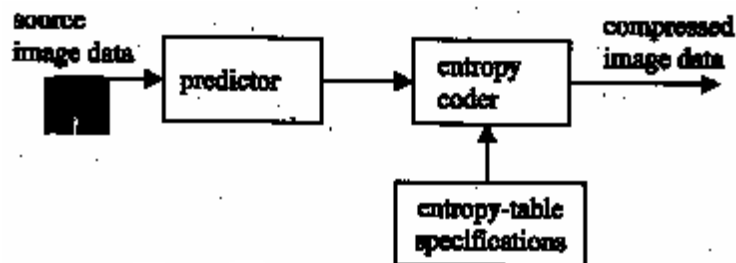
- Dapat digunakan untuk coding video (video = deretan gambar diam)
  - **Motion JPEG**
- Motion JPEG digunakan pada banyak aplikasi, terutama pada:
  - Video coding untuk transmisi melalui jaringan paket UBR, misalnya Internet (pada motion JPEG setiap frame dikodekan secara independen) → cocok untuk encoder video pada lingkungan yang *hostile*
  - Kompresi video untuk recording pada magnetic tapes → fleksibilitas editing, pause, fast forward, fast rewind, dll

## Mode Operasi JPEG

- JPEG mendefinisikan 4 mode operasi + varian:
  1. Sequential DCT-based mode
    - a. Baseline system
    - b. Extended sequential system
  2. Progressive DCT-based mode
    - a. Spectral Selection
    - b. Successive approximation
  3. Sequential lossless mode
  4. Hierarchical mode

## Sequential Lossless Mode

Menggunakan predictive coding

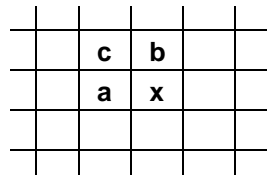


## Sequential Lossless Mode

Sumber image data dalam bentuk RGB atau  $Y C_b C_r$

Dalam bentuk sembarang format dengan sembarang ukuran dan presisi amplituda (mis. 8 bit/pixel)

- Predictor dalam bentuk DPCM
- Prediksi suatu input  $x$  didapat dari kombinasi sampai dengan tiga pixel tetangga  $a$ ,  $b$ ,  $c$  dari gambar yang sama pada komponen colour yang sama



## Sequential Lossless Mode

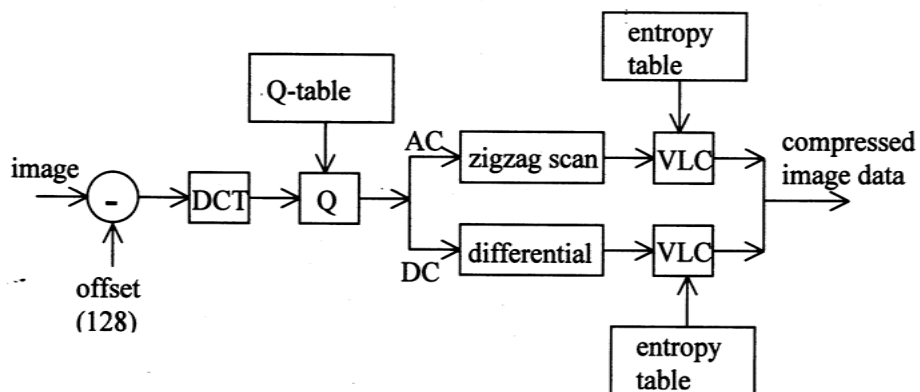
- Formula utk rekonstruksi sampel  $x$

Selection value	Prediction
0	no prediction
1	$x = a$
2	$x = b$
3	$x = c$
4	$x = a+b-c$
5	$x = a+(b-c)/2$
6	$x = b+(a-c)/2$
7	$x = (a+b)/2$

# Lossy Compression

- Standar JPEG mendefinisikan tiga mode lossy compression:
    - ☞ Sequential DCT-based mode
      - Baseline sequential mode → paling sederhana, kemampuannya cukup untuk banyak aplikasi
    - ☞ Progressive mode
    - ☞ Hierarchical mode
- Baseline decoding diperlukan menyediakan default decoding capability
- Semuanya berdasarkan DCT
  - Perbedaan mode-mode ini pada cara koefisien DCT ditransmisikan

## Baseline JPEG Encoder



## Baseline Sequential Mode Compression

⇒ Baseline Coding

- Image dipartisi dalam 8 × 8 blok pixel non-overlapping (dari kiri ke kanan atas ke bawah)
- Tiap blok dikodekan DCT
- 64 Koefisien transform dikuantisasi dengan kualitas yang diinginkan
- Koefisien hasil kuantisasi segera dikodekan entropy dan dikeluarkan sebagai output data kompresi

## Quantisation Table (Q-Table)

- Table didasarkan pada psychovisual thresholding
- Diturunkan secara empiris menggunakan luminance dan chrominance dengan 2 : 1 horizontal subsampling
- Tidak cocok untuk semua aplikasi tetapi memberikan hasil yang baik untuk kebanyakan image dengan presisi 8-bit

Luminance Q-Table

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Chrominance Q-Table

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

→ untuk  $q\_JPEG = 50$

## Kuantisasi

$$F^q(u, v) = \left\lfloor \frac{F(u, v)}{Q(u, v)} \right\rfloor$$

- $F^q(u, v)$  = koefisien transform setelah kuantisasi
- $F(u, v)$  = koefisien transform sebelum kuantisasi
- $Q(u, v)$  = elemen dr tabel kuantisasi luminance & chrominance
- Pada decoder inverse kuantisasi didapat dari:

$$F^Q(u, v) = F^q(u, v) \times Q(u, v)$$

## Kuantisasi

- Quality factor  $q\_JPEG$  digunakan untuk mengontrol elemen matrix kuantisasi  $Q(u, v)$
- Range dari harga  $q\_JPEG$  antara 1% - 100%
- Contoh matrix kuantisasi JPEG  $\rightarrow$  untuk  $q\_JPEG = 50$  (untuk luminance & chrominance)
- Untuk quality factor yang lain  $Q(u, v)$  dikalikan faktor kompresi  $\alpha$

$$\alpha = \frac{50}{q\_JPEG} \quad \text{jika } 1 \leq q\_JPEG \leq 50$$

$$\alpha = 2 - \frac{2 \times q\_JPEG}{100} \quad \text{jika } 50 \leq q\_JPEG \leq 99$$

- Untuk  $q\_JPEG = 100 \rightarrow \alpha \cdot Q(u, v) = 1$

## Contoh Kompresi JPEG



original pgm  
65674 bytes == 8 bpp



jpeg Q=50  
9420 bytes == 1.15 bpp



jpeg Q=25  
6388 bytes == 0.78 bpp

## Contoh Kompresi JPEG



original pgm  
65674 bytes == 8 bpp



jpeg Q=12  
4120 bytes == 0.5 bpp



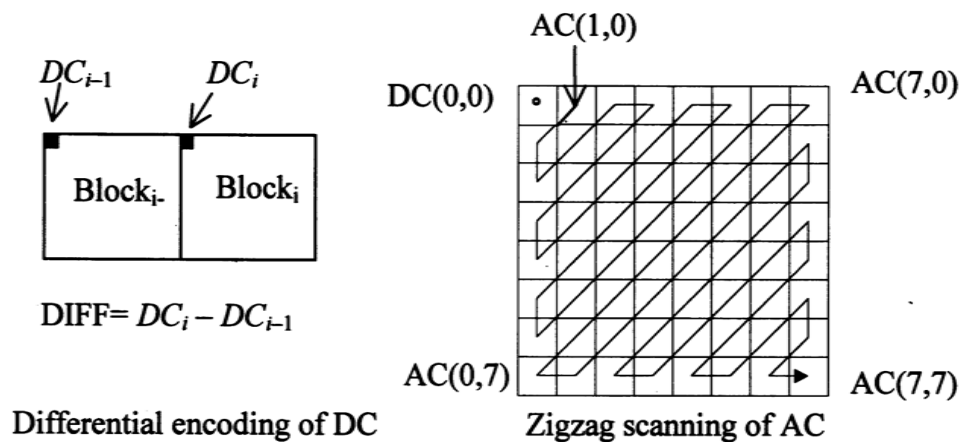
jpeg Q=5  
2209 bytes == 0.27 bpp



## Coding Untuk Koefisien DC dan AC

- Setelah kuantisasi DC (koefisien (0,0)) dan 63 koefisien AC dikodekan secara terpisah
- Koefisien DC dikodekan secara DPCM dengan prediksi dari koefisien DC blok sebelumnya ( $DIFF = DC_i - DC_{i-1}$ )
  - mengeksploitir korelasi antara harga DC dari blok berdekatan dan untuk pengkodekan lebih efisien
- 63 koefisien AC dimulai dari koefisien AC (1,0) dikodekan secara *Run-Length* mengikuti scan zigzag
  - mendapatkan deretan 0 yang panjang
  - untuk kebanyakan natural scene energi citra terutama terkonsentrasi pd beberapa koefisien transform energi rendah

## Menyiapkan Koefisien DCT untuk Entropy Coding



## Run-length Coding

Entropy coding untuk baseline encoder dilaksanakan dalam 2 tahap:

- VLC untuk bagian pertama → simbol-1
- Representasi biner amplituda → simbol-2

### CODING UNTUK KOEFISIEN DC

- Tiap DIFF tidak dialokasikan individual code-word
- Harga DIFF dikategorikan berdasarkan range magnitude → CAT
- CAT kemudian di dikodekan VLC → simbol-1
- CAT "ditambahkan" bit yang merepresentasikan simbol-2
- Karena harga koefisien DCT ada dalam range -2047 sampai 2047 → ada 11 kategori untuk koefisien yang tidak nol
- Kategori nol tidak digunakan untuk simbol tetapi untuk code end-of-block (EOB)

## CAT untuk Baseline Encoder

CAT	Range
0	-
1	-1,1
2	-3,-2,2,3
3	-7,...-4,4,...7
4	-15,...-8,8,...15
5	-31,...-16,16,...31
6	-63,...-32,32,...63
7	-127,...-64,64,...127
8	-255,...-128,128,...255
9	-511,...-256,256,...511
10	-1023,...-512,512,...1023
11	-2047,...-1024,1024,...2047

## Coding Untuk Koefisien DC

- Jika DIFF positif → bit yang 'ditambahkan' lower order bit dari DIFF
- Jika DIFF negatif → bit yang 'ditambahkan' lower order bit dari DIFF-1, dlm two complement representation
- Lower order bit mulai dari titik dimana most significant bit dari bit yang ditambahkan adalah 1 untuk DIFF positif dan 0 untuk DIFF negatif

Contoh 1:

DIFF = 6 → range 4 sampai 7 → CAT 3 → simbol-1 : 100  
bit yang ditambahkan DIFF=6=0000...00110 → simbol-2 : 110  
⇒ code keseluruhan untuk DIFF=6 adalah **100110**

Contoh 2:

DIFF = -3 → pada range -3 sampai -2 → CAT 2 → simbol-1 : 011  
bit yang ditambahkan DIFF-1=-4=1111..100 → simbol2: 100  
⇒ code keseluruhan untuk DIFF=-3 adalah **011100**

## Coding Untuk Koefisien AC

- Untuk tiap koef AC tidak nol pada scan zigzag → simbol-1 = 2-D event (RUN,CAT) biasa disebut juga RUN/SIZE
- CAT adalah kategori amplituda untuk koef tidak nol pada urutan zigzag
- RUN adalah jumlah nol sebelum koef tidak nol di atas
- Panjang maksimum run dibatasi 15
- Encoding untuk run > 15 dilakukan dengan simbol khusus (15,0)  
→ dapat sampai tiga simbol (15,0) berurutan sebelum simbol-1 diikuti simbol-2
- Contoh (RUN=34,CAT=5) menghasilkan 3 simbol a, b dan c, dengan a=(15,0), b=(15,0) dan c=(4,5)
- End-of-block (EOB) digunakan untuk sisa akhir koefisien pada urutan scanning zigzag dengan semua koefisien nol  
→ simbol EOB: (RUN=0,CAT=0)

## Coding Untuk Koefisien AC

- Tabel code AC untuk simbol-1 berisi satu code-word Huffman untuk tiap kemungkinan komposit event
- Format untuk bit tambahan (simbol-2) sama seperti coding DIFF pada koefisien DC
- Untuk koef AC ke  $k$  pada scan zigzag  $ZZ(k)$ , bit tambahan adalah lower order bit  $ZZ(k)$  jika  $ZZ(k)$  positif, atau lower order bit  $ZZ(k)-1$ , jika  $ZZ(k)$  adalah negatif

## Contoh Menghitung Code-Word

- Contoh : koefisien DCT setelah kuantisasi dari blok luminance diperlihatkan seperti dibawah. Koefisien DC blok sebelumnya adalah 29 → cari code-word untuk coding koef DC dan AC.

↓

31	18	0	0	0	0	0	0	0
-21	-13	0	0	0	0	0	0	0
0	5	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

## Contoh Menghitung Code-Word

Code-word untuk koefisien DC

- $\text{DIFF} = 31 - 29 = 2 \rightarrow \text{CAT} = 2 \rightarrow \text{Huffman code} : 011$
- $\text{DIFF} = 2 > 0$  maka  $2 = 000\dots0010$   $\rightarrow$  bit yang ditambahkan 10
- Code-word untuk koef DC = **01110**

Code-word untuk koefisien AC

- Scanning dari koef AC tidak nol yang pertama = 18  
 $\rightarrow \text{CAT} = 5 \rightarrow \text{RUN} = 0 \rightarrow \text{simbol-1} (0,5) \rightarrow \text{code-word } 11010$
- simbol-2 = lower order bit dari  $\text{ZZ}(k) = 18 = 000\dots010010$   
 $\rightarrow \text{simbol-2} = 10010$
- Code-word untuk koef AC pertama = **1101010010**

## Contoh Menghitung Code-Word

- Koef AC tidak nol berikutnya pada scan zigzag = -21  
 $\rightarrow \text{RUN} = 0, \text{CAT} = 5 \rightarrow \text{simbol-1} : 11010$
- Simbol-2 :  $21 < 0$ , maka  $\text{ZZ}(k)-1 = -22 = 111\dots1101010$   
 $\rightarrow \text{simbol-2} = 01010$
- Code-word untuk koef AC kedua = **1101001010**
  
- Koef AC tidak nol ketiga pada scan zigzag = -13  
 $\rightarrow \text{RUN} = 1, \text{CAT} = 4 \rightarrow \text{simbol-1} : 11110110$
- Simbol-2:  $\text{ZZ}(k)-1 = -14 = 111\dots110010$   
 $\rightarrow \text{simbol-2} = 0010$
- Code-word untuk koef AC ketiga = **111101100010**

## Contoh Menghitung Code-Word

- Koef AC tidak nol keempat pada scan zigzag = 5  
→ RUN = 3, CAT = 3 → simbol-1 : 11111110101
- Simbol-2:  $ZZ(k) = 5 = 000\dots00\underline{101}$   
→ simbol-2 = 101
- Code-word untuk koef AC keempat = **111111110101101**
- Karena 5 adalah koef AC tidak nol terakhir  
→ encoding berakhir disini dan code EOB  
ditransmisikan → simbol (0,0) tanpa bit tambahan →  
code-word 1010

## Entropy Coding

- Untuk coding magnitude categories atau event *runlength* standar JPEG menspesifikasikan 2 alternatif metoda entropy coding:
  - Huffman coding → Huffman Table (entropy table specifications)
  - Arithmetic coding → arithmetic coding conditioning tables (entropy table specifications)
- Ada sampai 4 tabel Huffman dan Arithmetic coding yang berbeda untuk masing-masing koefisien DC dan AC
- Tidak ada harga default yang dispesifikasikan untuk tabel Huffman
  - aplikasi dapat memilih tabel yang sesuai dengan lingkungannya

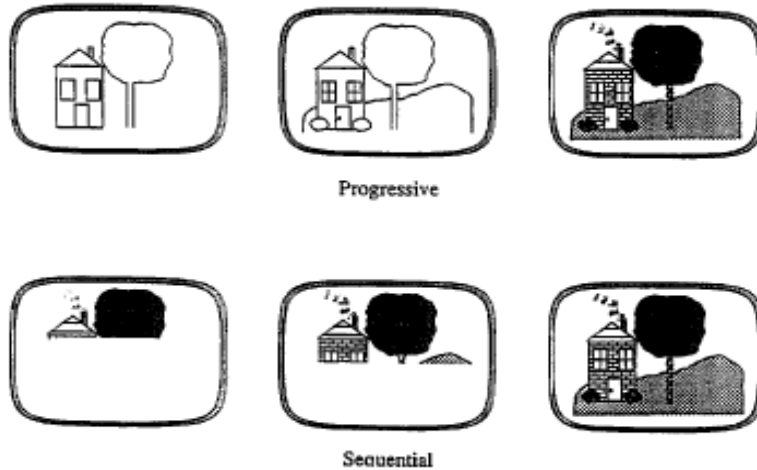
## Entropy Coding

- Default tabel didefinisikan untuk arithmetic coding conditioning
- Baseline sequential coding menggunakan Huffman Coding
- Extended DCT-based and lossless processes dapat menggunakan Huffman atau Arithmetic Coding

## Extended Sequential System

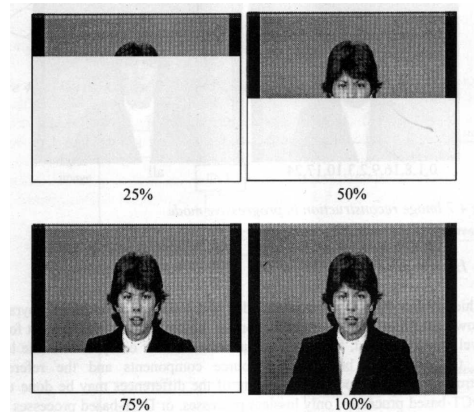
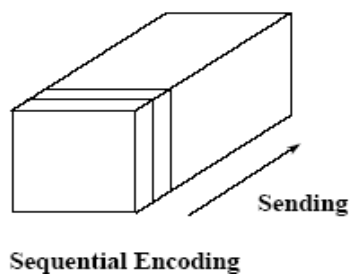
- Baseline encoder
  - hanya mendukung basic coding tools
    - mencukupi untuk kebanyakan aplikasi kompresi image
  - Input image dengan presisi pixel 8-bit
  - Huffman coding untuk *runlength*
  - Sequential transmission
- Extended DCT-based process
  - sembarang presisi image diperlukan
  - digunakan arithmetic coding untuk kompresi yang lebih tinggi

## Sequential vs Progressive Mode



## Sequential Mode

- Satu blok pixel dikodekan  $\rightarrow$  64 koef-nya dikuantisasi dikodekan & ditransmisikan
- Pada penerima setelah decoding koefisien  $\rightarrow$  inverse quantisation, inverse transformation secara sekuensial merekonstruksi image
- Waktu rekonstruksi keseluruhan image tergantung rate kanal.

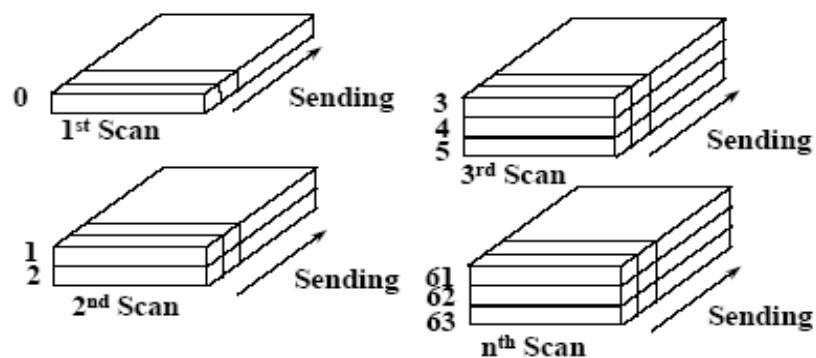




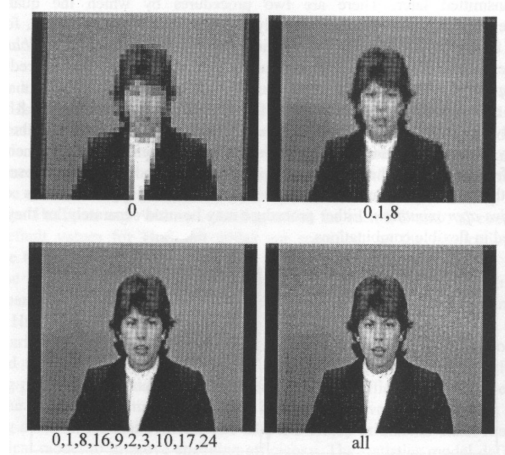
# Progressive Mode

- Koefisien terkuantisasi disimpan dalam buffer lokal → ditransmisikan kemudian
- Ada 2 prosedur isi buffer diencode secara partial dalam satu scan:
  - Spectral selection : pertama-tama hanya band (koefisien) tertentu yang dikodekan  
→ koefisien-koefisien lainnya selanjutnya
  - Successive approximation : Encoding pertama dari koefisien hanya most significant bit tertentu yang dikodekan  
→ less significant bit dikodekan berikutnya
- Masing-masing prosedur dapat digunakan secara terpisah atau dicampur dalam kombinasi yang fleksibel

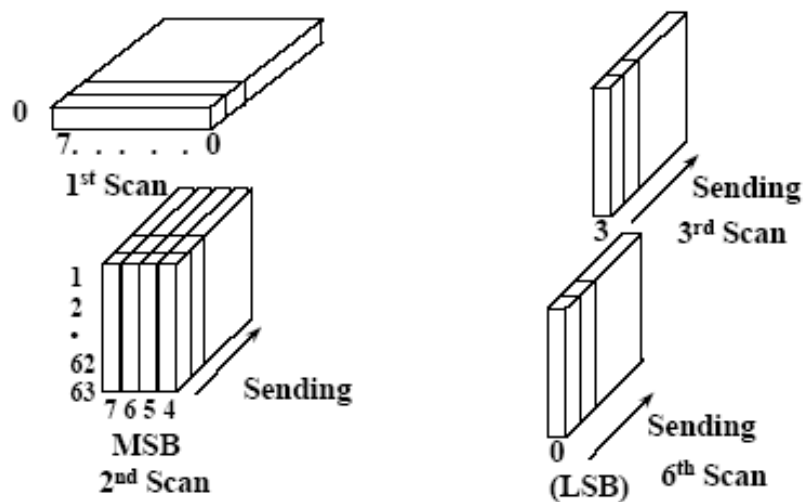
## Progressive Mode: Spectral Selection



## Contoh Progressive Mode : Spectral Selection



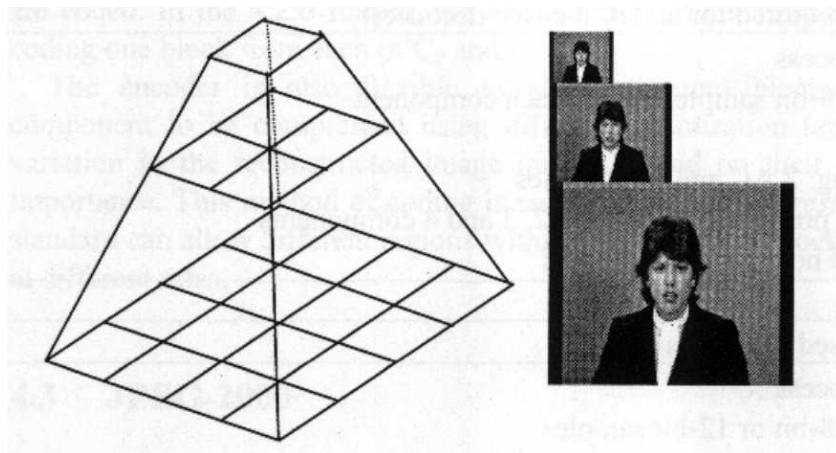
## Progressive Mode: Successive Approximation



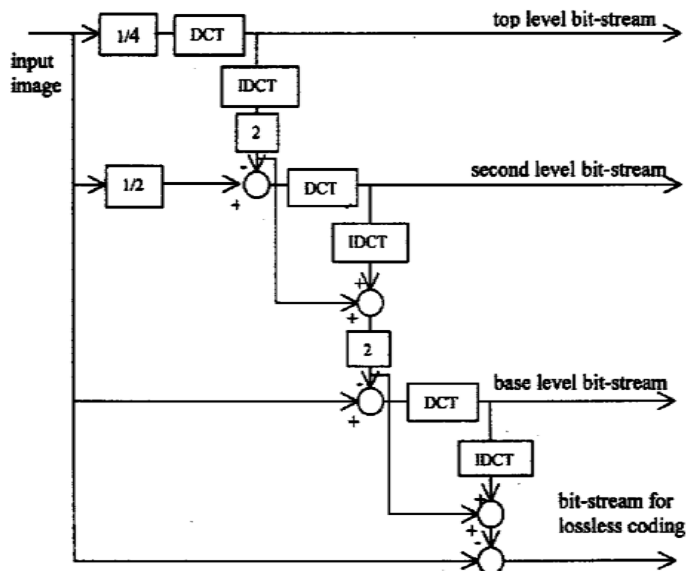
## Hierarchical Mode

- Image dikodekan sebagai deretan layer sebuah piramid
- Tiap image dengan ukuran yang lebih kecil menyediakan prediksi untuk layer berikutnya yang lebih atas
- Kecuali untuk top level pada piramid, pada lower level, perbedaan antara source component dan reference-reconstructed image dikodekan (untuk tiap luminance dan colour component)
- Coding perbedaan dapat dilakukan menggunakan hanya proses DCT-based, proses lossless atau process DCT-based dan final proses lossless
- Downsampling dan upsampling filter dapat digunakan untuk menyediakan resolusi spatial dari piramid
- Berguna utk transmisi citra resolusi tinggi ke penerima resolusi rendah

## Hierarchical Multiresolution Encoding



# Three-level Hierarchical Encoder



## Baseline process (required for all DCT-based decoders)

- DCT-based process
- Source image: 8-bit samples within each component
- Sequential
- Huffman coding: 2 AC and 2 DC tables
- Decoders shall process scans with 1, 2, 3 and 4 components
- Interleaved and non-interleaved scans

## Extended DCT-based processes

- DCT-based process
- Source image: 8-bit or 12-bit samples
- Sequential or progressive
- Huffman or Arithmetic coding: 4 AC and 4 DC tables
- Decoder shall process scans with 1, 2, 3 and 4 components
- Interleaved and non-interleaved scans

## Lossless process

- Predictive process (not DCT-based)
- Source image;  $N$ -bit samples ( $2 \leq N \leq 16$ )
- Sequential
- Huffman or Arithmetic coding: 4 tables
- Decoders shall process scans with 1, 2, 3 and 4 components
- Interleaved and non-interleaved scans

## Hierarchical processes

- Multiple layers (non-differential and differential)
- Uses extended DCT-based or lossless processes
- Decoders shall process scans with 1, 2, 3 and 4 components
- Interleaved and non-interleaved scans

(RUN,CAT)	Code-word	(RUN,CAT)	Code-word	(RUN,CAT)	Code-word	(RUN,CAT)	Code-word
0,0 (EOB)	1010	4,1	111011	8,1	111111000	12,1	1111111010
0,1	00	4,2	111111000	8,2	11111111000000	12,2	111111111011001
0,2	01	4,3	111111110010110	8,3	111111110110110	12,3	111111111011010
0,3	100	4,4	111111110010111	8,4	111111110110111	12,4	111111111011011
0,4	1011	4,5	111111110011000	8,5	111111110110111	12,5	111111111011100
0,5	11010	4,6	111111110011001	8,6	111111110110100	12,6	111111111011101
0,6	1111000	4,7	111111110011010	8,7	111111110110110	12,7	111111111011110
0,7	11111000	4,8	111111110011011	8,8	111111110110111	12,8	111111111011111
0,8	1111110110	4,9	111111110011100	8,9	111111110111100	12,9	111111111100000
0,9	111111110000010	4,10	111111110011101	8,10	111111110111101	12,10	111111111100001
0,10	111111110000011	5,1	1111010	9,1	111111001	13,1	11111111000
1,1	1100	5,2	11111110111	9,2	111111110111110	13,2	111111111100010
1,2	1101	5,3	111111110011110	9,3	111111110111111	13,3	111111111100011
1,3	1111001	5,4	111111110011111	9,4	111111110000000	13,4	111111111100100
1,4	111110110	5,5	111111110100000	9,5	111111111000001	13,5	111111111100101
1,5	1111110110	5,6	111111110100001	9,6	111111111000010	13,6	111111111100110
1,6	111111110000100	5,7	111111110100010	9,7	111111111000011	13,7	111111111100111
1,7	111111110000101	5,8	111111110100011	9,8	111111111000100	13,8	111111111101000
1,8	111111110000110	5,9	111111110100100	9,9	111111111000101	13,9	111111111101001
1,9	111111110000111	5,10	111111110100101	9,10	111111111000110	13,10	111111111101010
1,10	111111110001000	6,1	1111011	10,1	111111010	14,1	111111111101011
2,1	11100	6,2	111111110110	10,2	111111111000111	14,2	111111111101100
2,2	11111001	6,3	111111110100110	10,3	111111111001000	14,3	111111111101101
2,3	1111110111	6,4	111111110100111	10,4	111111111001001	14,4	111111111101110
2,4	111111110100	6,5	111111110101000	10,5	111111111001010	14,5	111111111101111
2,5	111111110001001	6,6	111111110101001	10,6	111111111001011	14,6	111111111100000
2,6	111111110001010	6,7	111111110101010	10,7	111111111001011	14,7	111111111100001
2,7	111111110001011	6,8	111111110101011	10,8	111111111001100	14,8	111111111100010
2,8	111111110001100	6,9	111111110101100	10,9	111111111001110	14,9	111111111100011
2,9	111111110001101	6,10	111111110101101	10,10	111111111001111	14,10	111111111101000
2,10	111111110001110	7,1	11111010	11,1	111111001	15,1	111111111101001
3,1	111010	7,2	111111110111	11,2	111111111010000	15,2	111111111101010
3,2	111110111	7,3	111111110101110	11,3	111111111010001	15,3	111111111101011
3,3	111111110101	7,4	111111110101111	11,4	111111111010010	15,4	111111111101100
3,4	111111110001111	7,5	111111110110000	11,5	111111111010011	15,5	111111111101101
3,5	111111110001000	7,6	111111110110001	11,6	111111111010010	15,6	111111111101110
3,6	111111110001001	7,7	111111110110010	11,7	111111111010101	15,7	111111111101111
3,7	111111110001010	7,8	111111110110011	11,8	111111111010110	15,8	111111111101110
3,8	111111110001011	7,9	111111110110100	11,9	111111111010111	15,9	111111111101111
3,9	111111110001100	7,10	111111110110101	11,10	1111111101000	15,10	111111111101110
3,10	111111110001101						111111111101111

The special symbol representing 16 "zero" 15,0 (ZRL)