

# PEMAMPATAN CITRA

*(Image Compression)*

# Pemampatan vs Pengkodean

- Pemampatan
  - Citra dikodekan
  - Representasi Memory menjadi lebih kecil
  - Menerapkan proses Compress dan Decompress
  - Aplikasi : Pengiriman dan Penyimpanan Data
- Pengkodean
  - Citra dikodekan
  - Representasi Memory belum tentu lebih kecil
  - Menerapkan proses Encode dan Decode

# Kriteria Pemampatan

- Waktu pemampatan
- Kebutuhan memory
- Kualitas pemampatan (*fidelity*)

$$PSNR = 20 \times \log_{10} \left( \frac{b}{rms} \right)$$

$$rms = \sqrt{\frac{1}{\text{Lebar} \times \text{Tinggi}} \sum_{i=1}^N \sum_{j=1}^M (f_{ij} - f'_{ij})^2}$$

- Format Keluaran

# Jenis Pemampatan

- Pendekatan Statistik
  - Melihat frekuensi kemunculan derajat keabuan pixel
- Pendekatan Ruang
  - Melihat hubungan antar pixel yang mempunyai derajat keabuan yang sama pada wilayah dalam citra
- Pendekatan Kuantisasi
  - Mengurangi jumlah derajat keabuan yang tersedia
- Pendekatan Fraktal
  - Kemiripan bagian citra dieksploitasi dengan matriks transformasi

# Klasifikasi Metode Pemampatan

- Metode Lossless

- menghasilkan citra yang sama dengan citra semula
- Tidak ada informasi yang hilang
- Nisbah/ratio pemampatan sangat rendah
- Contoh, metode Huffman

$$\text{Nisbah} = 100\% - \left( \frac{\text{ukuran citra hasil pemampatan}}{\text{ukuran citra semula}} \times 100\% \right)$$

- Metode lossy

- menghasilkan citra yang *hampir* sama dengan citra semula
- Ada informasi yang hilang akibat pemampatan tapi masih bisa ditolerir oleh persepsi mata
- Nisbah/ratio pemampatan tinggi
- Contoh, JPEG dan Fraktal



Sebelum



Sesudah

# Metode Pemampatan Huffman

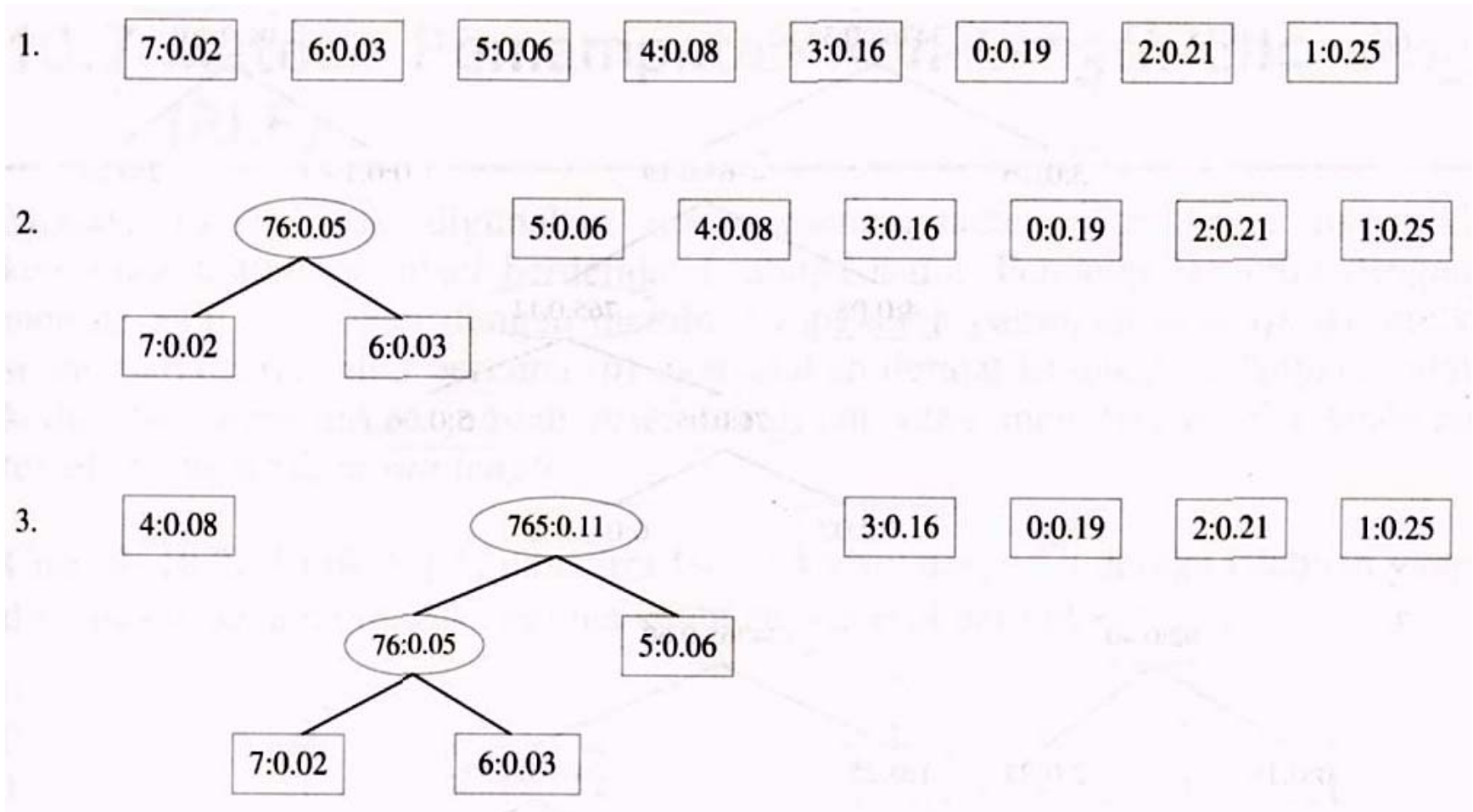
1. Urutkan nilai keabuan berdasarkan frekuensi kemunculannya
2. Gabung dua pohon yang frekuensi kemunculannya paling kecil
3. Ulangi 2 langkah diatas sampai tersisa satu pohon biner
4. Beri **label 0** untuk pohon **sisi kiri** dan **1** untuk pohon **sisi kanan**
5. Telusuri barisan label sisi dari akar ke daun yang menyatakan kode Huffman

# Metode Pemampatan Huffman

- Contoh, citra 64x64 dengan 8 derajat keabuan ( $k$ )

$k$	$n_k$	$p(k) = n_k/n$
0	790	0.19
1	1023	0.25
2	850	0.21
3	656	0.16
4	329	0.08
5	245	0.06
6	122	0.03
7	81	0.02

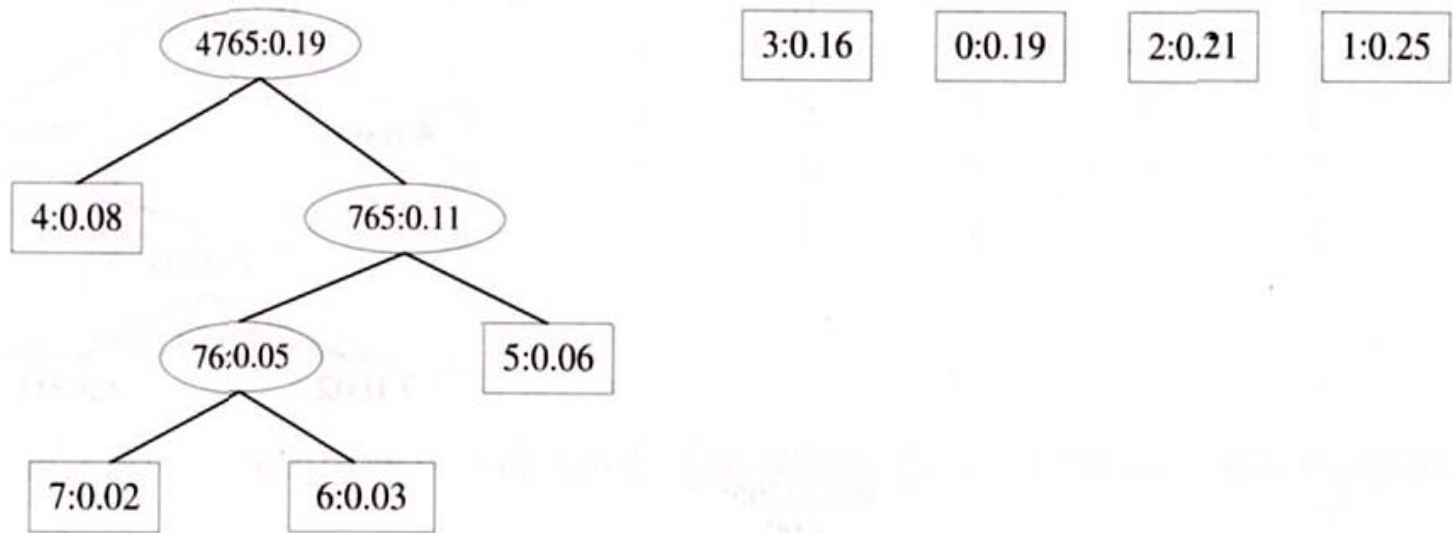
# Metode Pemampatan Huffman





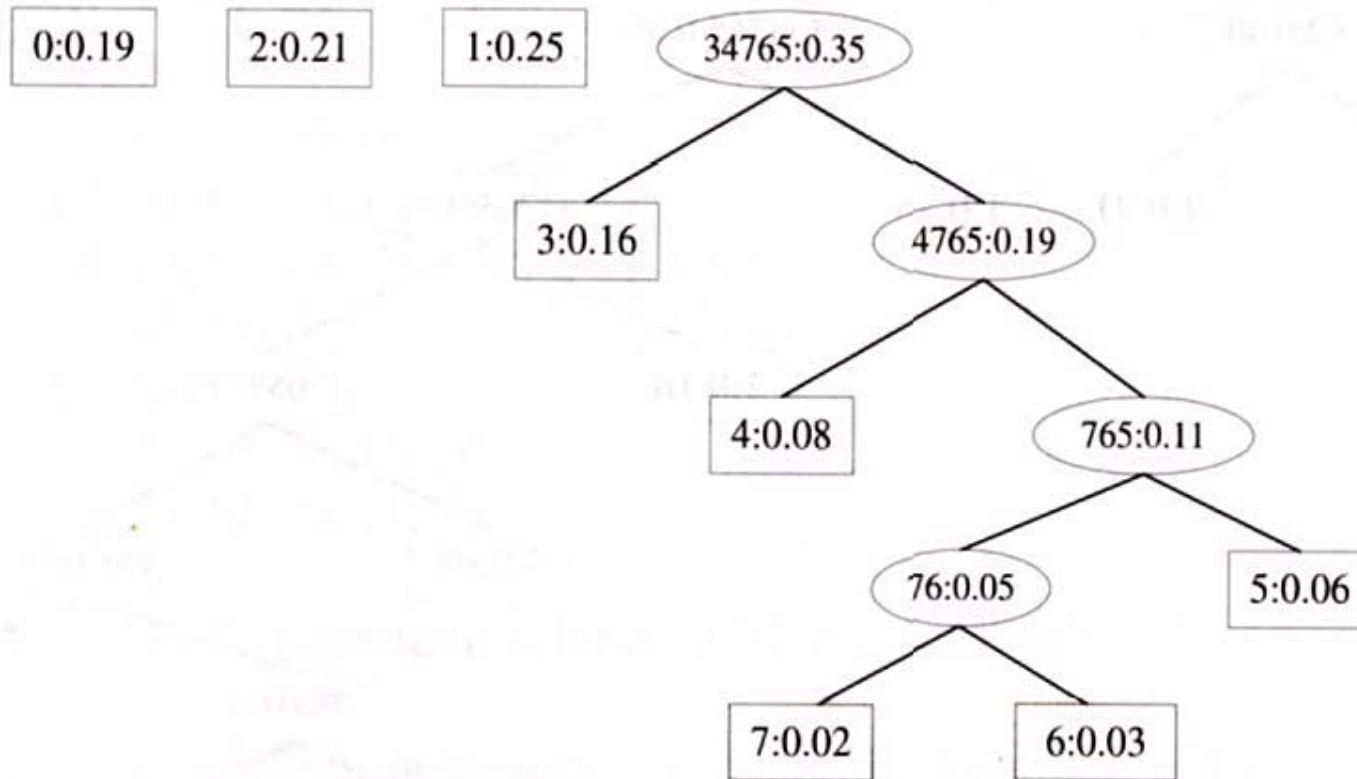
# Metode Pemampatan Huffman

4.



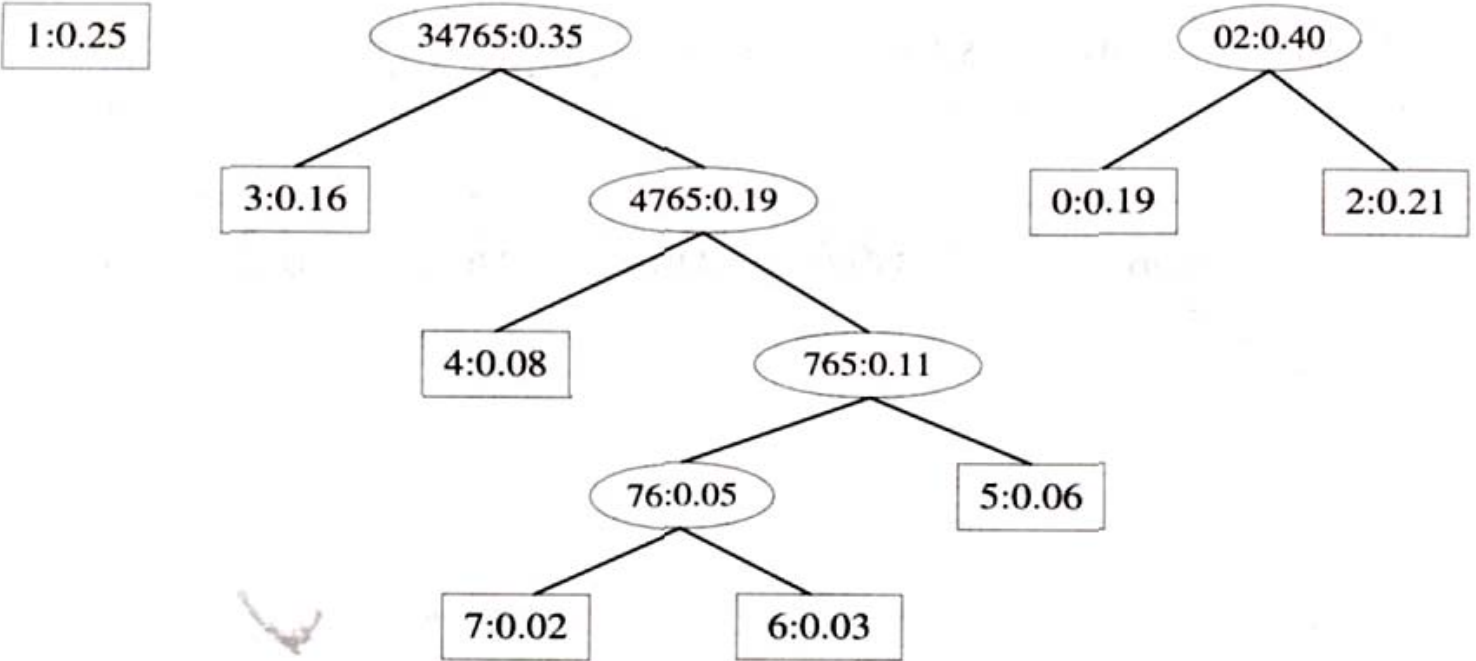
# Metode Pemampatan Huffman

5.

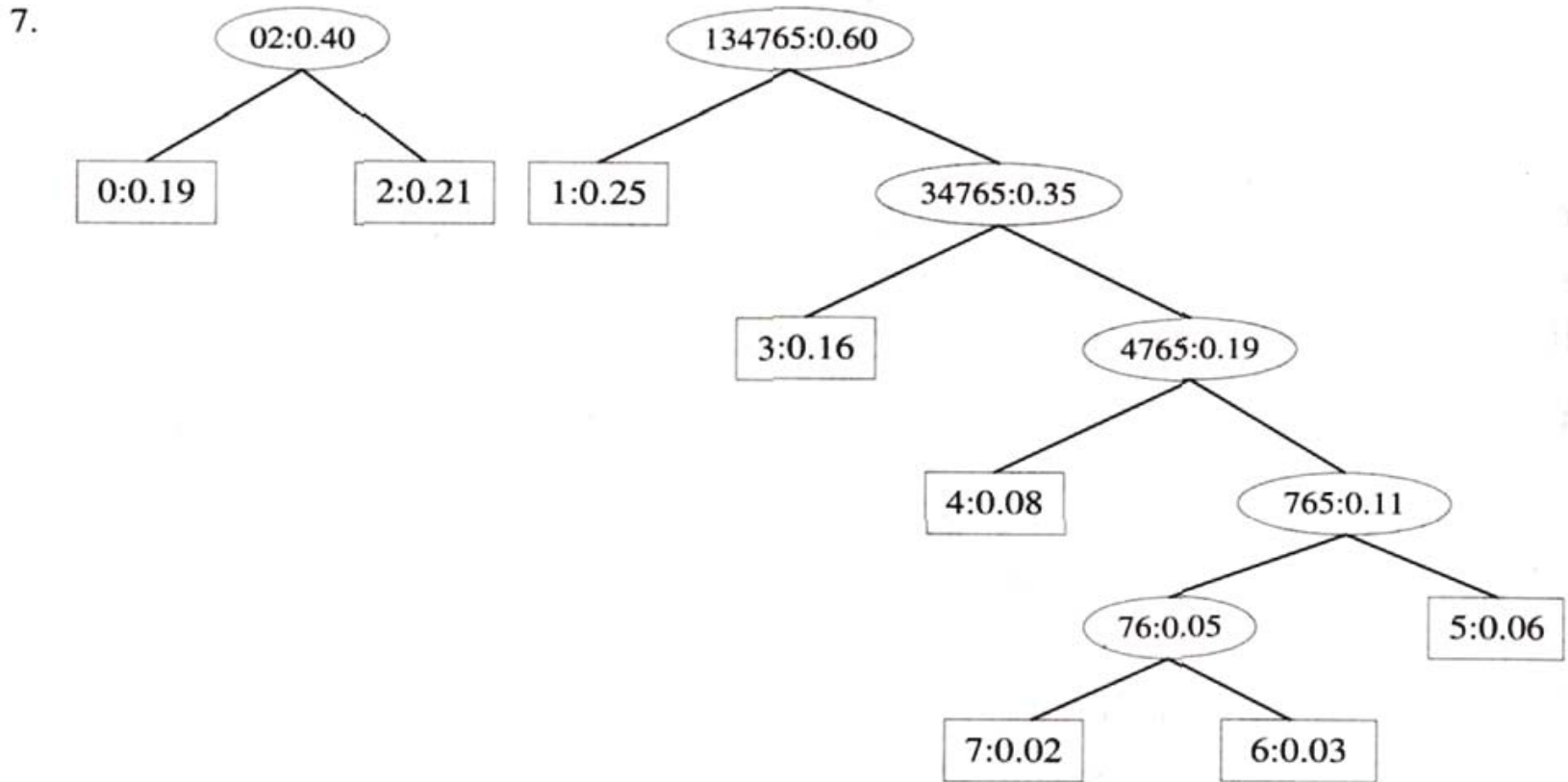


# Metode Pemampatan Huffman

6.

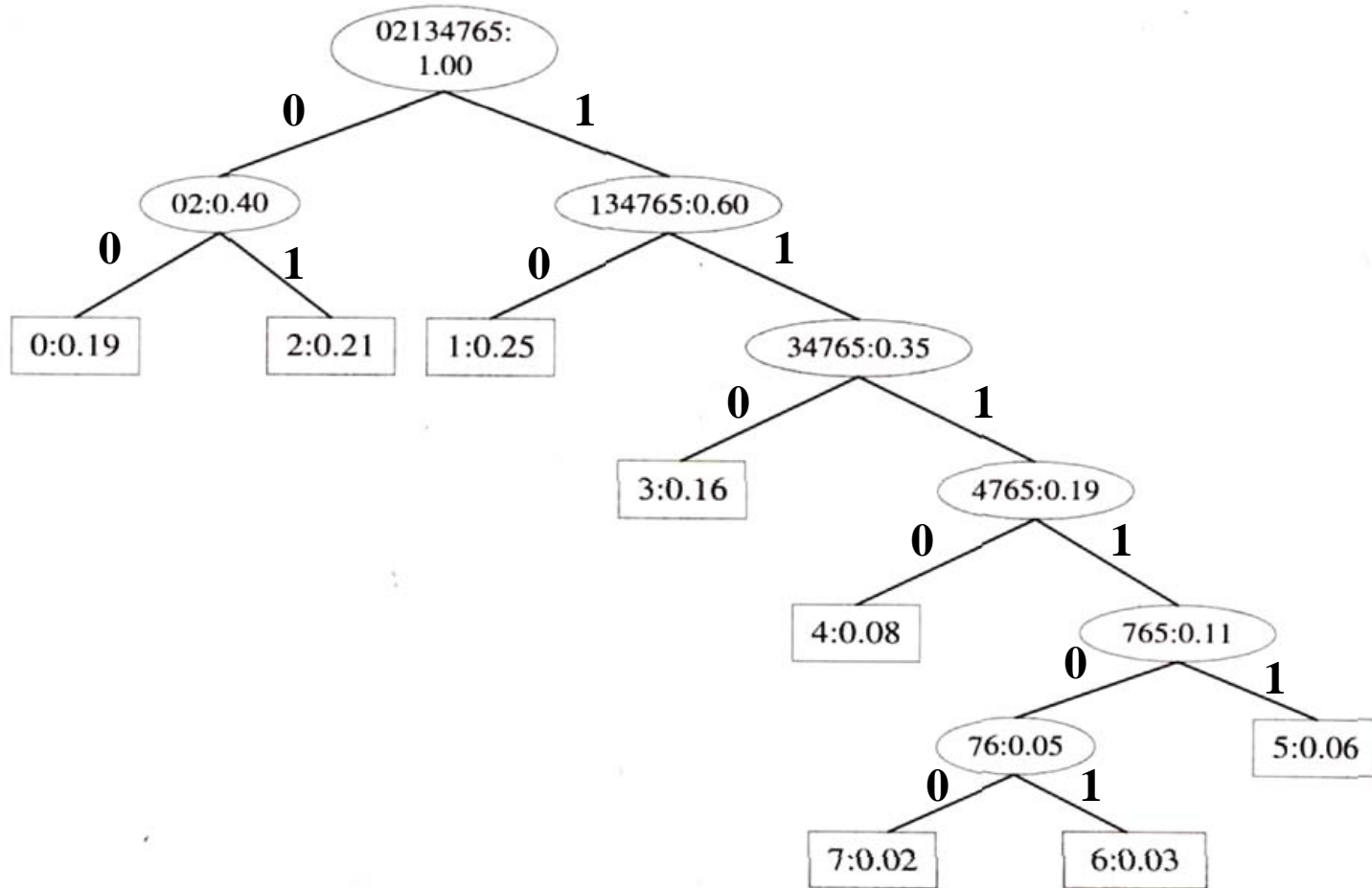


# Metode Pemampatan Huffman



# Metode Pemampatan Huffman

8.



# Metode Pemampatan Huffman

- Contoh, citra 64x64 dengan 8 derajat keabuan ( $k$ )
- Kode untuk setiap derajat keabuan

0 = 00

2 = 01

4 = 1110

6 = 111101

1 = 10

3 = 110

5 = 11111

7 = 111100

- Ukuran citra sebelum dimampatkan (1 derajat keabuan = 3 bit) adalah  $4096 \times 3 \text{ bit} = 12288 \text{ bit}$
- Ukuran citra setelah pemampatan

$(790 \times 2 \text{ bit}) + (1023 \times 2 \text{ bit}) + (850 \times 2 \text{ bit}) +$

$(656 \times 3 \text{ bit}) + (329 \times 4 \text{ bit}) + (245 \times 5 \text{ bit}) +$

$(122 \times 6 \text{ bit}) + (81 \times 6 \text{ bit}) = 11053 \text{ bit}$

Nisbah pemampatan =  $(100\% - \frac{11053}{12288} \times 100\%) = 10\%$

# Metode Pemampatan RLE

- **Run Length Encoding**

- Cocok untuk pemampatan citra yang memiliki kelompok pixel berderajat keabuan yang sama

- Contoh citra 10x10 dengan 8 derajat keabuan

0	0	0	0	0	2	2	2	2	2	(0, 5), (2, 5)
0	0	0	1	1	1	1	2	2	2	(0, 3), (1, 4), (2, 3)
1	1	1	1	1	1	1	1	1	1	(1, 10)
4	4	4	4	3	3	3	3	2	2	(4, 4), (3, 4), (2, 2)
3	3	3	5	5	7	7	7	7	6	(3, 3), (5, 2), (7, 4), (6, 1)
2	2	6	0	0	0	0	1	1	0	(2, 2), (6, 1), (0, 4), (1, 2), (0, 1)
3	3	4	4	3	2	2	2	1	1	(3, 2), (4, 2), (3, 1), (2, 2), (1, 2)
0	0	0	0	0	0	0	0	1	1	(0, 8), (1, 2)
1	1	1	1	0	0	0	2	2	2	(1, 4), (0, 3), (2, 3)
3	3	3	2	2	2	1	1	1	1	(3, 3), (2, 3), (1, 4)

Pasangan derajat keabuan ( $p$ )  
dan jumlah pixel ( $q$ )

# Metode Pemampatan RLE

- Ukuran citra sebelum dimampatkan (1 derajat keabuan = 3 bit) adalah  $100 \times 3 \text{ bit} = 300 \text{ bit}$
- Ukuran citra setelah pemampatan (run length =4) adalah  $(31 \times 3) + (31 \times 4) \text{ bit} = 217 \text{ bit}$

Nisbah pemampatan =  $(100\% - \frac{217}{300} \times 100\%) = 27.67\%$ , yang artinya 27.67%



# Metode Pemampatan Kuantisasi

- Buat histogram citra yang akan dimampatkan.  $P$  jumlah pixel
- Identifikasi  $n$  buah kelompok di histogram sedemikian sehingga setiap kelompok mempunyai kira-kira  $P/n$  pixel
- Nyatakan setiap kelompok dengan derajat keabuan 0 sampai  $n-1$ . Setiap kelompok dikodekan kembali dengan nilai derajat keabuan yang baru

# Metode Pemampatan Kuantisasi

- Contoh, Citra 5 x 13

```

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
    
```

- Akan dimampatkan dengan 4 derajat keabuan (0 - 3) atau dengan 2 bit

## Histogram

```

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

## Kelompoknya

13	0 **	0
	1 **	
	2 *****	
20	3 *****	
	4 *****	1
	5 ****	
17	6 *****	2
	7 *****	
15	8 *****	3
	9 *****	

# Metode Pemampatan Kuantisasi

- Setelah dimampatkan

0	3	2	1	3	0	2	1	3	2	3	1	2
1	3	2	1	2	2	1	3	0	3	1	2	1
1	3	1	2	1	3	0	1	3	0	2	1	3
1	3	1	2	0	2	2	0	0	2	2	1	0
0	0	1	1	3	3	2	1	2	0	0	3	0

- Ukuran sebelum pemampatan (1 derajat keabuan = 4 bit) adalah  $65 \times 4 \text{ bit} = 260 \text{ bit}$
- Ukuran citra setelah pemampatan (1 derajat keabuan = 2 bit) adalah  $65 \times 2 \text{ bit} = 130 \text{ bit}$

$$\text{Nisbah pemampatan} = (100\% - \frac{130}{260} \times 100\%) = 50\%$$