

# Konsep Dasar Citra Digital

Hari ke-2

Dra. Hernawati, M.T

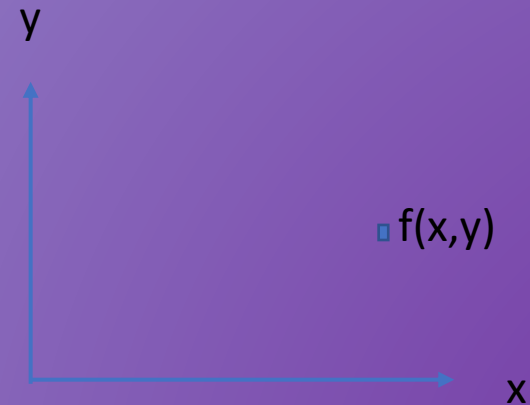
# Pembentukan Citra

# Model Citra

- Secara matematis fungsi intensitas cahaya pada bidang dwimatra disimbolkan dengan  $f(x, y)$ , yang dalam hal ini:

$(x, y)$  : koordinat pada bidang dwimatra

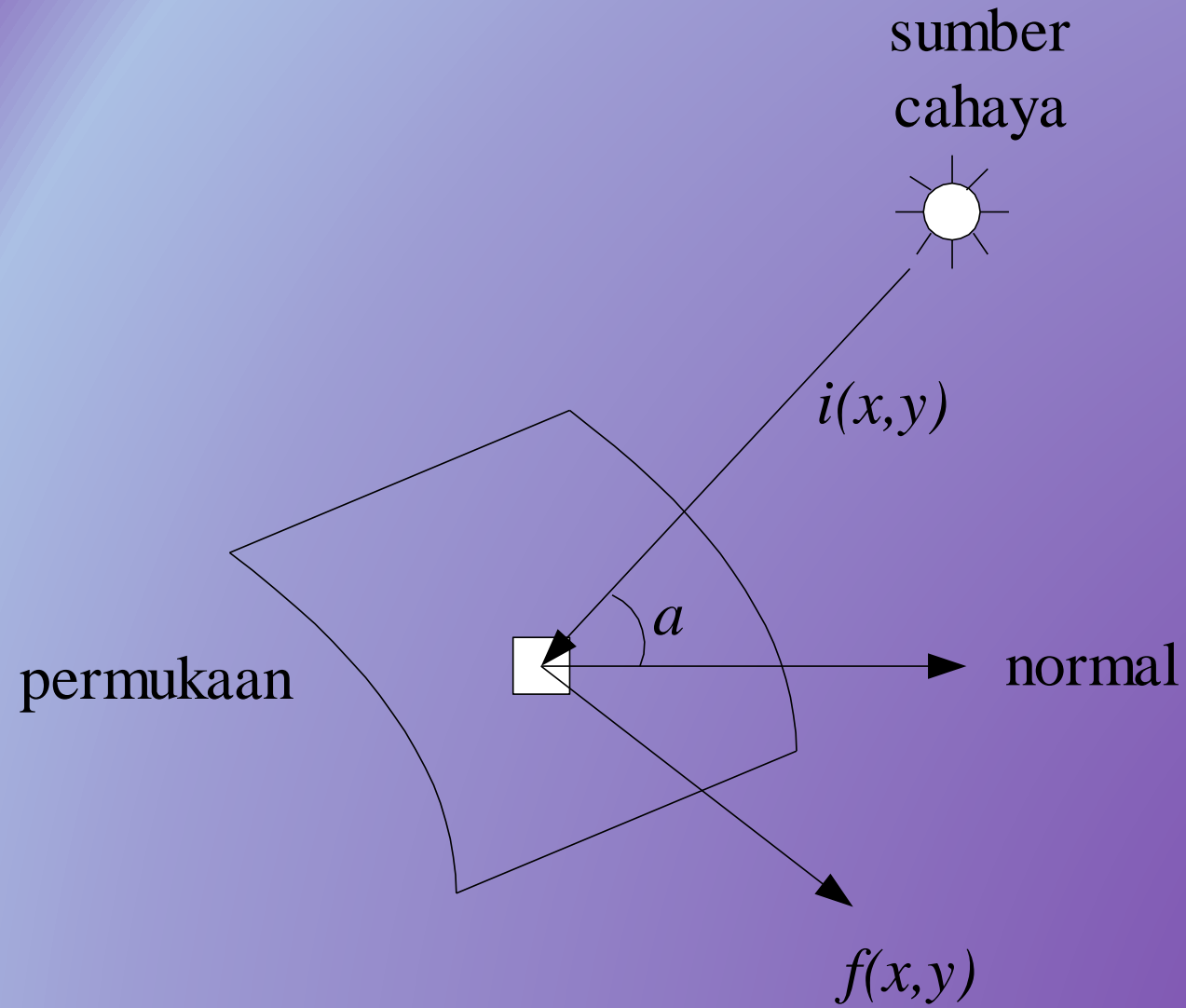
$f(x, y)$  : intensitas cahaya (*brightness*)  
pada titik  $(x, y)$



- Karena cahaya merupakan bentuk energi, maka intensitas cahaya bernilai antara 0 sampai tidak berhingga,

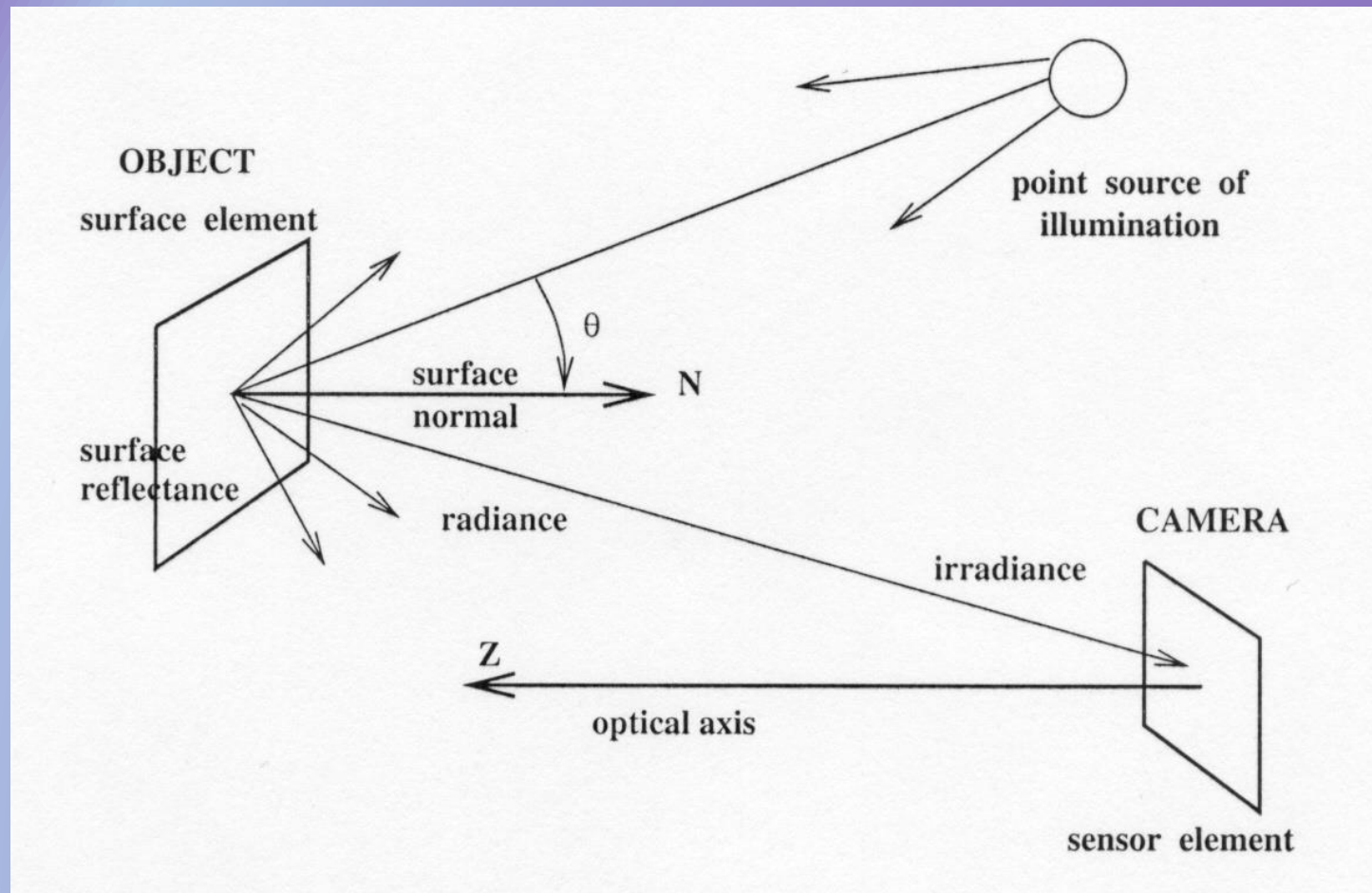
$$0 \leq f(x, y) < \infty$$

- Nilai  $f(x, y)$  sebenarnya adalah hasil kali:  
 $i(x, y)$  = jumlah cahaya yang berasal dari sumbernya (*illumination*),  
nilainya antara 0 sampai tidak berhingga, dan  
 $r(x, y)$  = derajat kemampuan obyek memantulkan cahaya  
(*reflection*), nilainya antara 0 dan 1.
- Jadi,  $f(x, y) = i(x, y) \cdot r(x, y)$ ,  
yang dalam hal ini,  
$$0 \leq i(x, y) < \infty$$
$$0 \leq r(x, y) \leq 1$$



Sinyal  $f(x,y)$  ini yang ditangkap oleh mata atau kamera

The scene is illuminated by a single source.  
The scene reflects radiation towards the camera.  
The camera senses it via chemicals on film.



- Nilai  $i(x, y)$  ditentukan oleh sumber cahaya, sedangkan  $r(x, y)$  ditentukan oleh karakteristik objek di dalam gambar.
- Nilai  $r(x, y) = 0$  mengindikasikan penyerapan total, sedangkan  $r(x, y) = 1$  menyatakan pemantulan total.
- Jika permukaan mempunyai derajat pemantulan nol, maka fungsi intensitas cahaya,  $f(x, y)$ , juga nol.
- Sebaliknya, jika permukaan mempunyai derajat pemantulan 1, maka fungsi intensitas cahaya sama dengan iluminasi yang diterima oleh permukaan tersebut.

Contoh-contoh nilai  $i(x, y)$ :

- pada hari cerah, matahari menghasilkan iluminasi  $i(x, y) \approx 9000 \text{ foot candles}$ ,
- pada hari mendung (berawan), matahari menghasilkan iluminasi  $i(x, y) \approx 1000 \text{ foot candles}$ ,
- pada malam bulan purnama, sinar bulan menghasilkan iluminasi  $i(x, y) \approx 0.01 \text{ foot candle}$ .

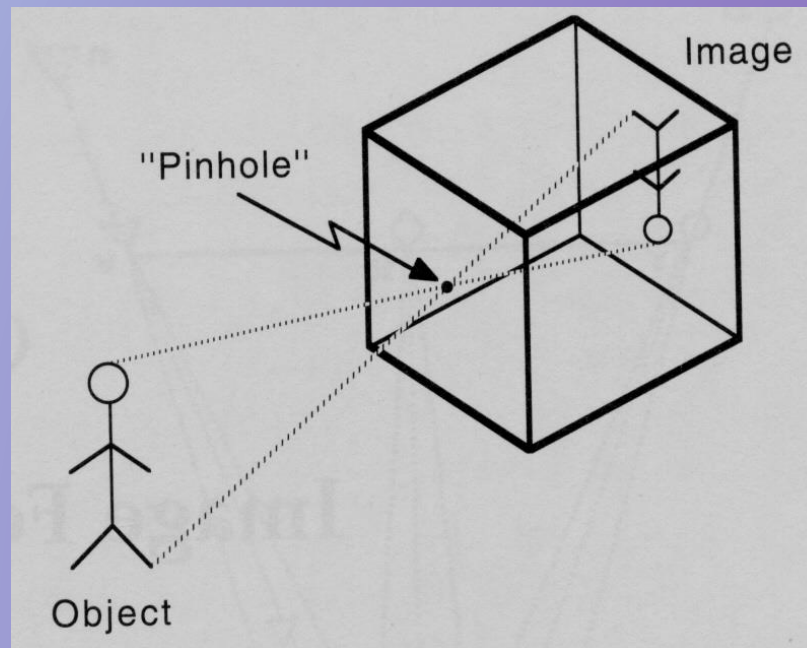
Contoh nilai  $r(x, y)$

- benda hitam mempunyai  $r(x, y) = 0.01$ ,
- dinding putih mempunyai  $r(x, y) = 0.8$ ,
- benda logam dari *stainlesssteel* mempunyai  $r(x, y) = 0.65$ ,
- salju mempunyai  $r(x, y) = 0.93$ .



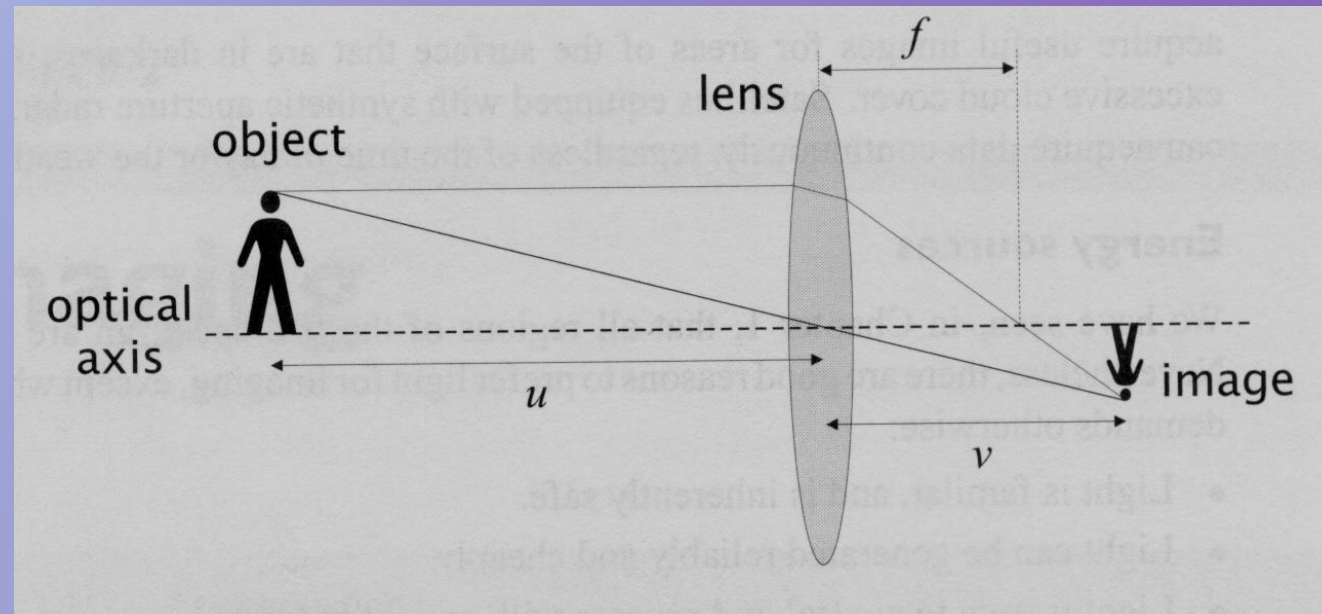
# Pinhole camera

- This is the simplest device to form an image of a 3D scene on a 2D surface.
- Straight rays of light pass through a “pinhole” and form an inverted image of the object on the image plane.



# Camera optics

- In practice, the aperture must be larger to admit more light.
- Lenses are placed to in the aperture to **focus** the bundle of rays from each scene point onto the corresponding point in the image plane

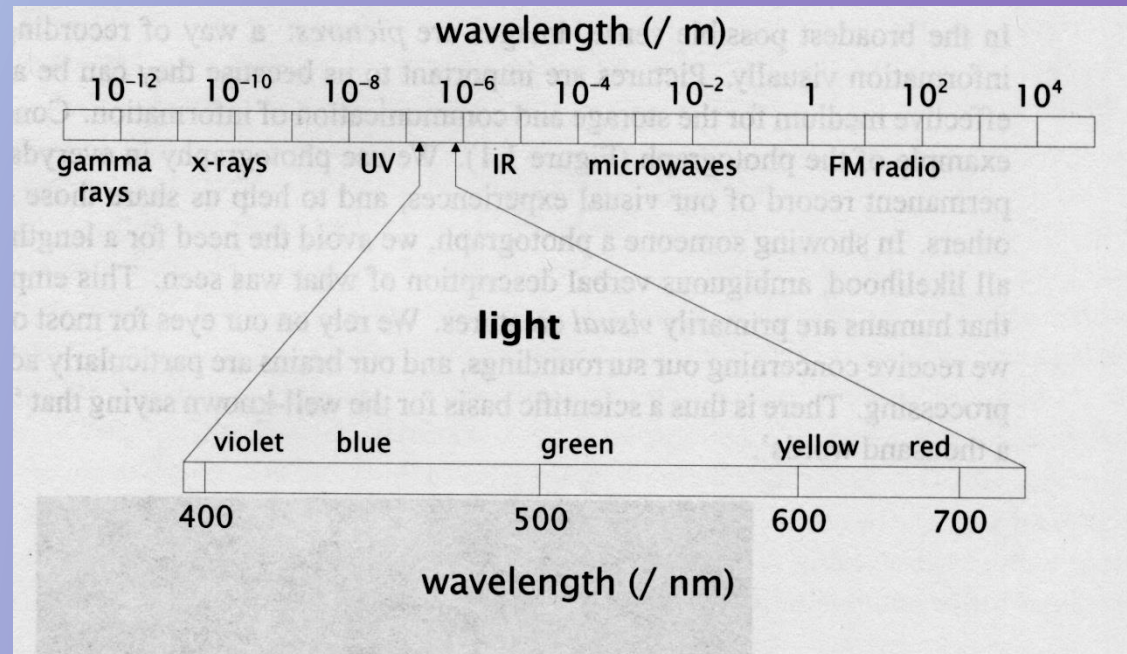


# Image formation

- Optical parameters of the lens
  - lens type
  - focal length
  - field of view
- Photometric parameters
  - type, intensity, and direction of illumination
  - reflectance properties of the viewed surfaces
- Geometric parameters
  - type of projections
  - position and orientation of camera in space
  - perspective distortions introduced by the imaging process

# What is light?

- The visible portion of the **electromagnetic** (EM) spectrum.
- It occurs between wavelengths of approximately 400 and 700 nanometers.



# Short wavelengths

- Different wavelengths of radiation have different properties.
- The **x-ray** region of the spectrum, it carries sufficient energy to penetrate a significant volume or material.





# Long wavelengths

- Copious quantities of **infrared** (IR) radiation are emitted from warm objects (e.g., locate people in total darkness).



# Sonic images

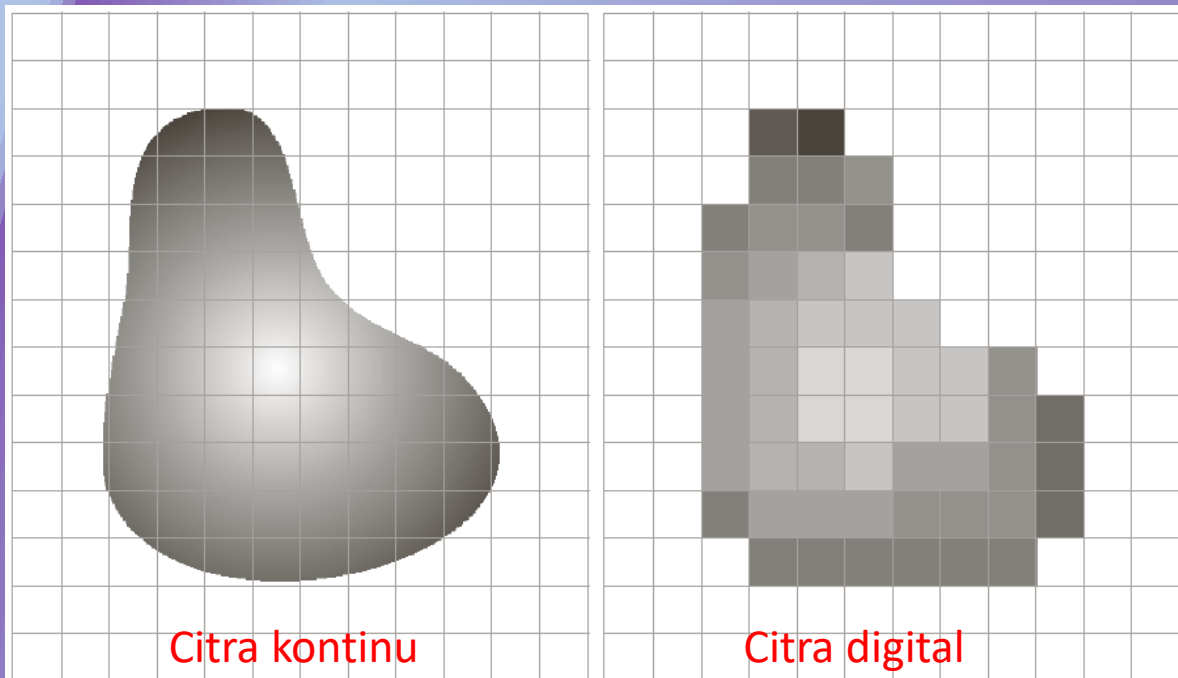
- Produced by the reflection of sound waves off an object.
- High sound frequencies are used to improve resolution.



# Digitalisasi Citra



- Citra adalah sinyal kontinu dwimatra,  $f(x, y)$
- $f(x, y)$  menyatakan intensitas cahaya pada posisi  $(x, y)$
- Agar citra dapat diolah oleh komputer digital, maka citra perlu di-digitalisasi (atau di-digitisasi) menjadi citra digital



a b

**FIGURE 2.17** (a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.

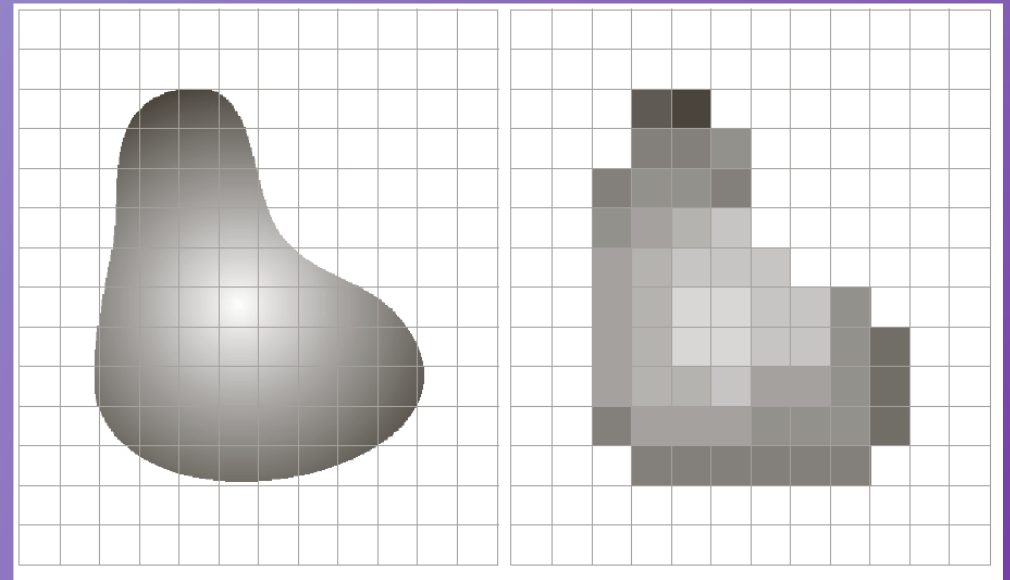
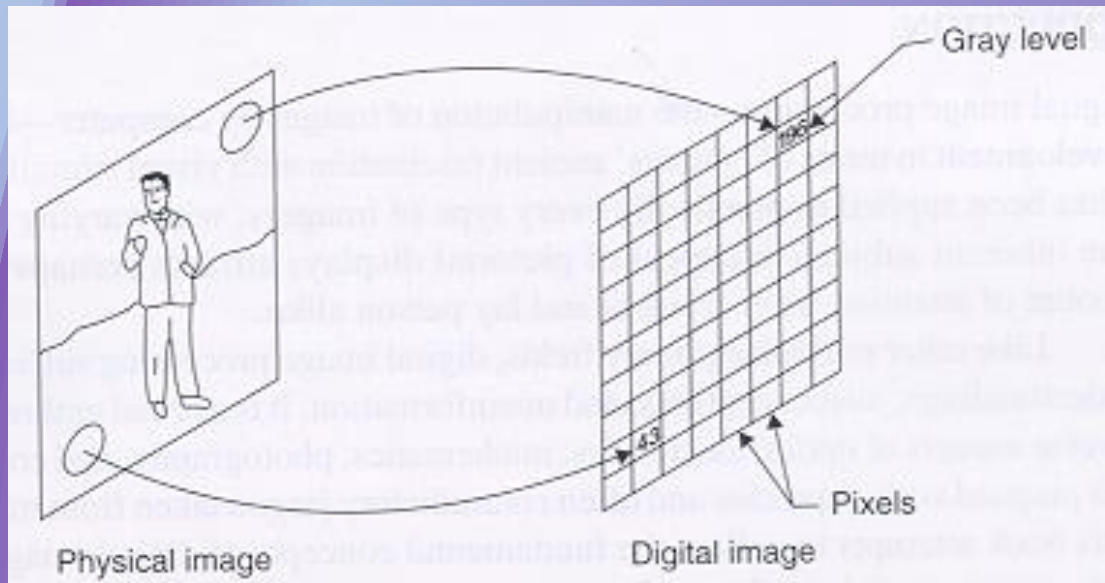
Sebuah citra digital adalah versi diskrit dari citra kontinu

- Citra digital  $f(x, y)$  direpresentasikan sebagai matriks berukuran  $N \times M$

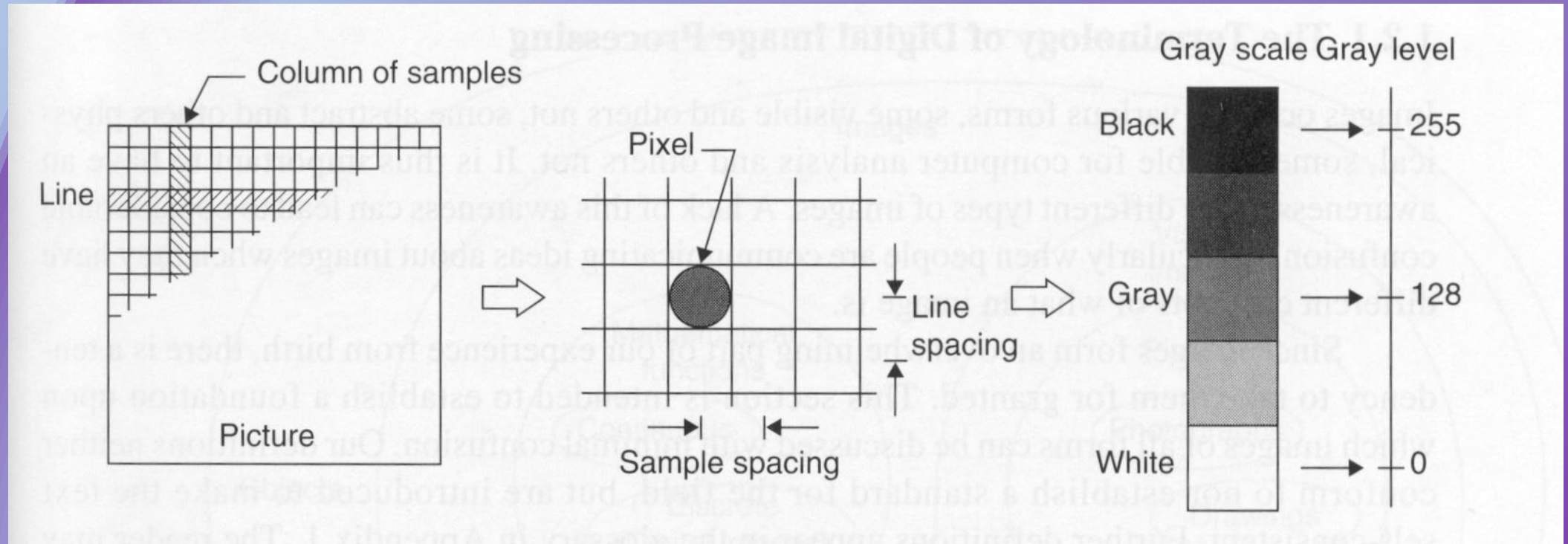
$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \dots & \dots & \dots & \dots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix}$$

- $N \times M$  menyatakan resolusi citra,  $N$  baris dan  $M$  kolom
- Setiap elemen matriks menyatakan sebuah *pixel* (*picture element*)
- Nilai  $f(i, j)$  menyatakan nilai intensitas *pixel* pada posisi  $(i, j)$ , yang dinamakan *graylevel* (derajat keabuan).

- Proses digitalisasi citra ada dua tahap:
  1. Penerokan (*sampling*): yaitu digitalisasi secara spasial ( $x, y$ ).
  2. Kuantisasi: yaitu pengangkaan nilai intensitas  $f(x, y)$  menjadi *integer*.
- Kedua proses di atas berkaitan dengan *diskritisasi* tetapi dalam ranah berbeda

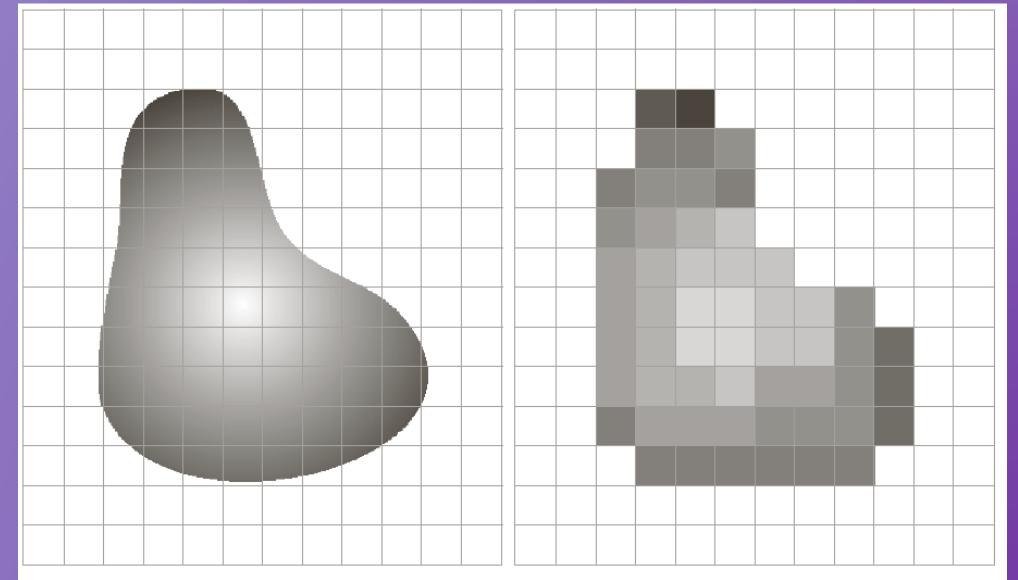
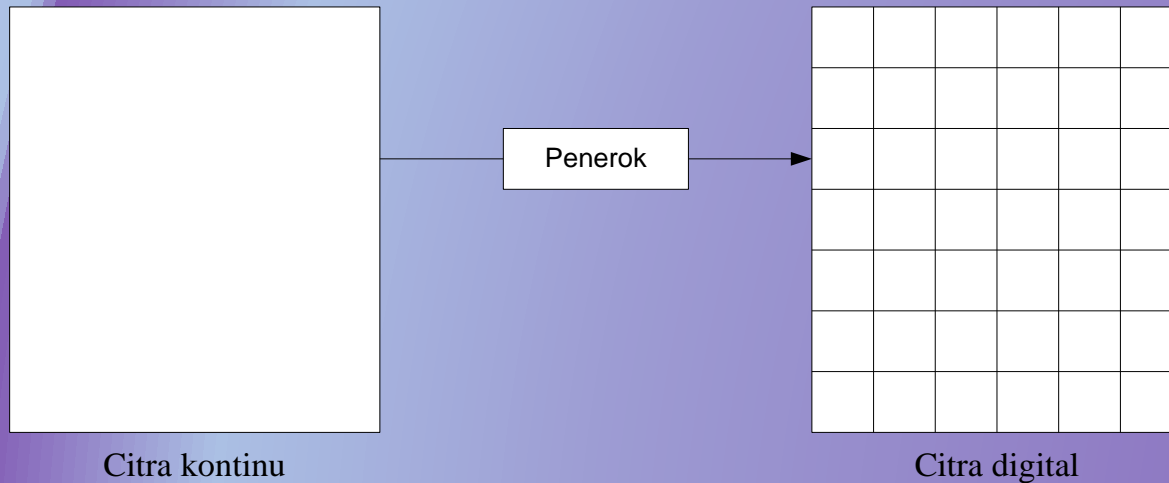


## Proses digitalisasi citra

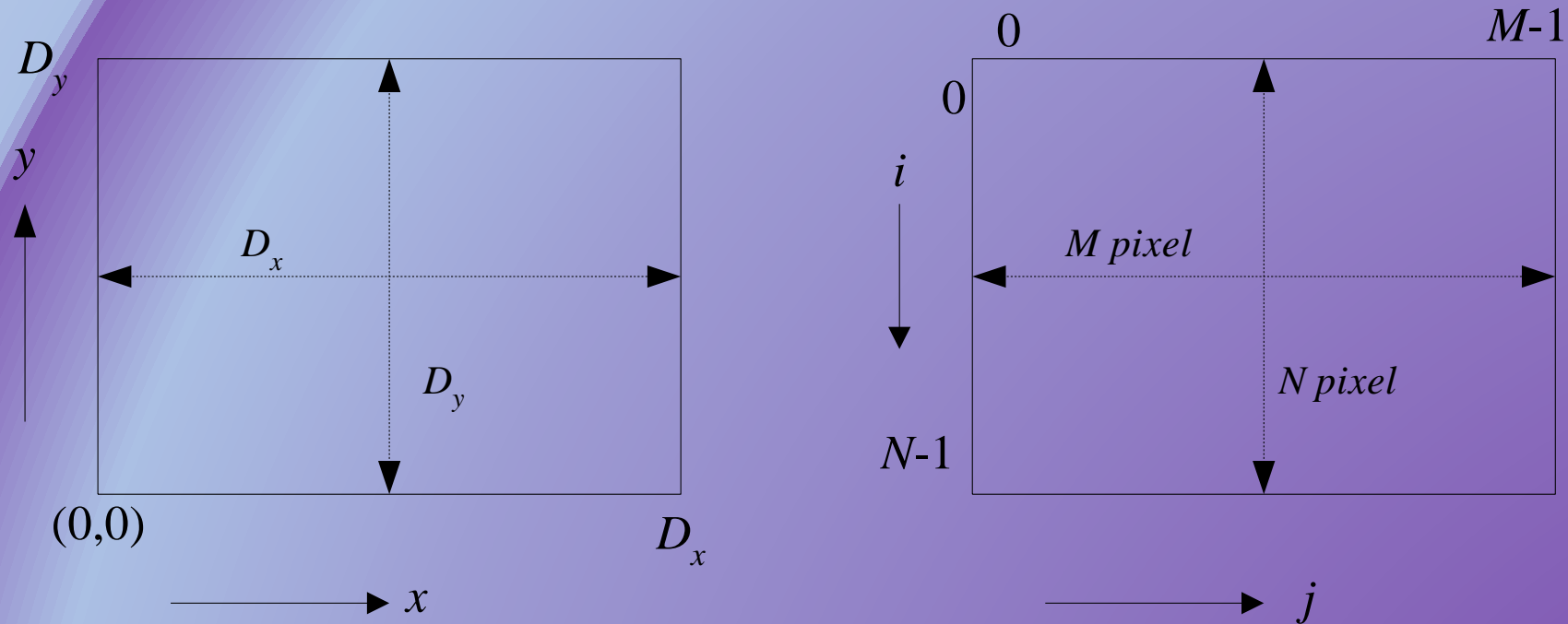


# sampling

- Citra kontinu diambil menjadi grid-grid yang berbentuk bujursangkar
- sampling bertujuan untuk menentukan seberapa banyak *pixel* yang diperlukan untuk merepresentasikan citra kontinu, dan bagaimana pengaturannya



## Hubungan antara elemen gambar dan elemen matriks



$$0 \leq i \leq N - 1$$

$$0 \leq j \leq M - 1$$

$N$  = jumlah maksimum *pixel* dalam satu baris

$M$  = jumlah maksimum *pixel* dalam satu kolom

$D_x$  = lebar gambar (dalam inchi)

$D_y$  = tinggi gambar (dalam inchi)



- Jumlah hasil sampling biasanya diasumsikan perpangkatan dari dua,

$$N = 2^n$$

yang dalam hal ini,

$N$  = jumlah sampling pada suatu baris/kolom

$n$  = bilangan bulat positif

- Contoh ukuran sampling:  $256 \times 256$  *pixel*,  
 $128 \times 256$  *pixel*,  
 $512 \times 1024$  *pixel*  
dst

## Contoh sampling:

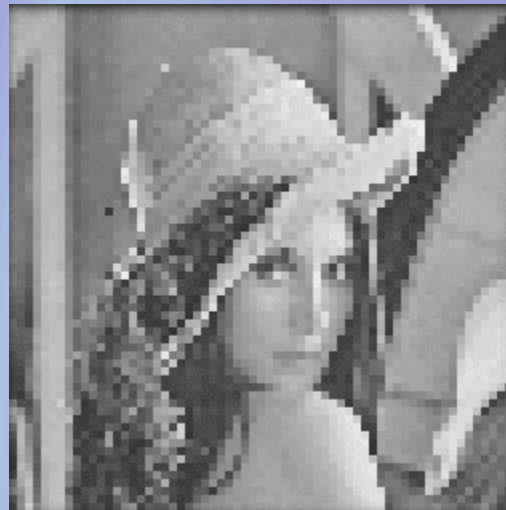
original image,  $n = 8$  (256 x 256)



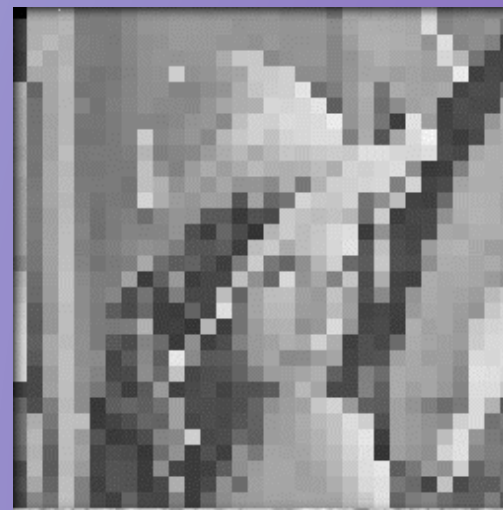
$n = 7$  (128 x 128)



$n = 6$  (64 x 64)



$n = 5$  (32 x 32)

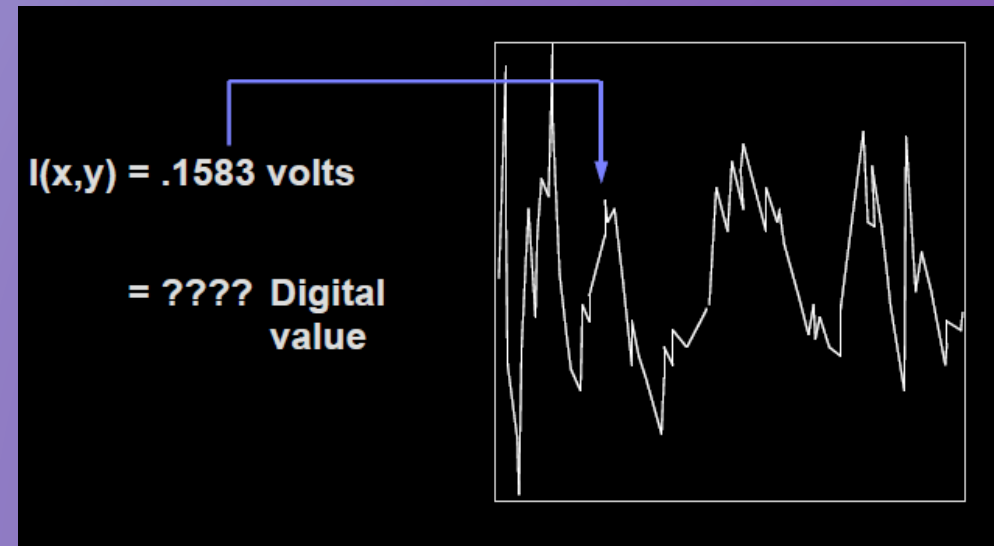
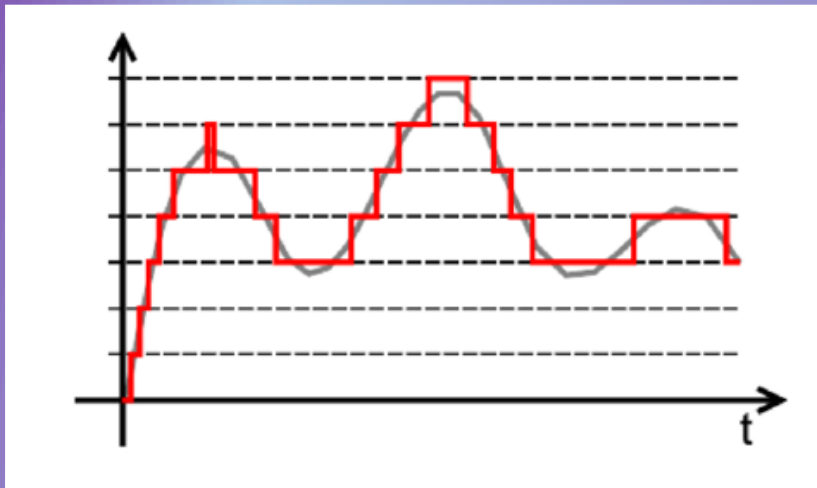


Catatan: Citra telah  
disamakan ukurannya  
untuk perbandingan



# Kuantisasi

- Kuantisasi berkaitan dengan diskritisasi nilai intensitas cahaya pada koordinat (x, y).
- Tujuan kuantisasi adalah memetakan nilai dari sinyal kontinu menjadi G buah nilai diskrit



- Nilai intensitas *pixel* dalam integer dinyatakan dalam selang  $[0, L]$ .
- Selang  $[0, L]$  disebut skala keabuan (*graylevel*)
- Proses kuantisasi membagi skala keabuan  $[0, L]$  menjadi  $G$  buah level.
- Biasanya  $G$  diambil perpangkatan dari 2,

$$G = 2^m$$

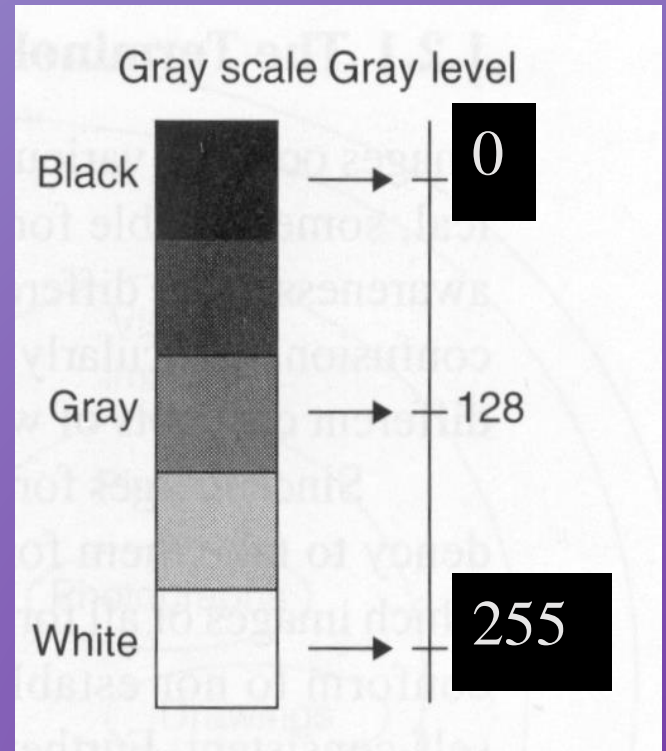
yang dalam hal ini,

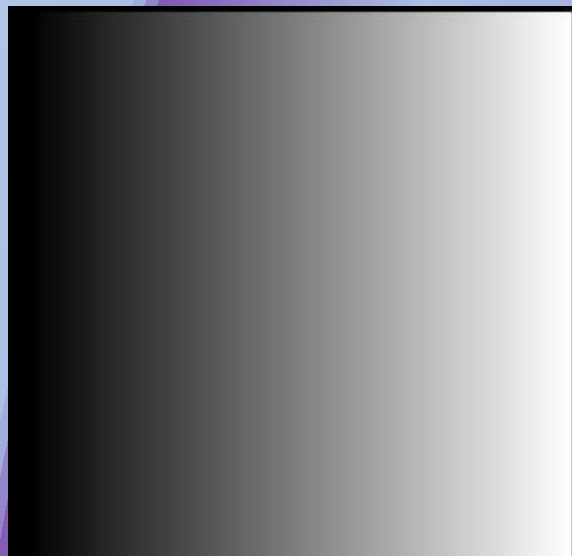
$G$  = derajat keabuan

$m$  = bilangan bulat positif

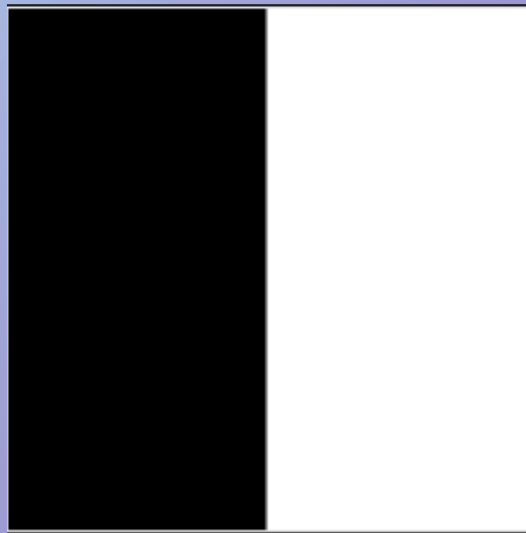
$$G = 2^m$$

| Skala Keabuan     | Rentang Nilai Keabuan | <i>Pixel Depth</i> |
|-------------------|-----------------------|--------------------|
| $2^1$ (2 nilai)   | 0, 1                  | 1 bit              |
| $2^2$ (4 nilai)   | 0 sampai 3            | 2 bit              |
| $2^3$ (16 nilai)  | 0 sampai 15           | 3 bit              |
| $2^8$ (256 nilai) | 0 sampai 255          | 8 bit              |





Original image



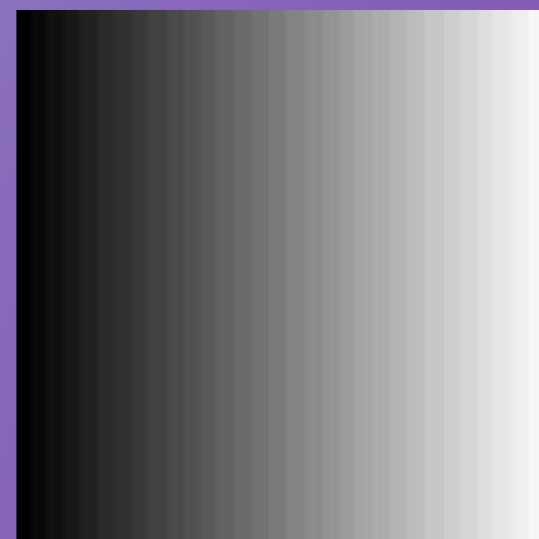
2 graylevel



4 graylevel

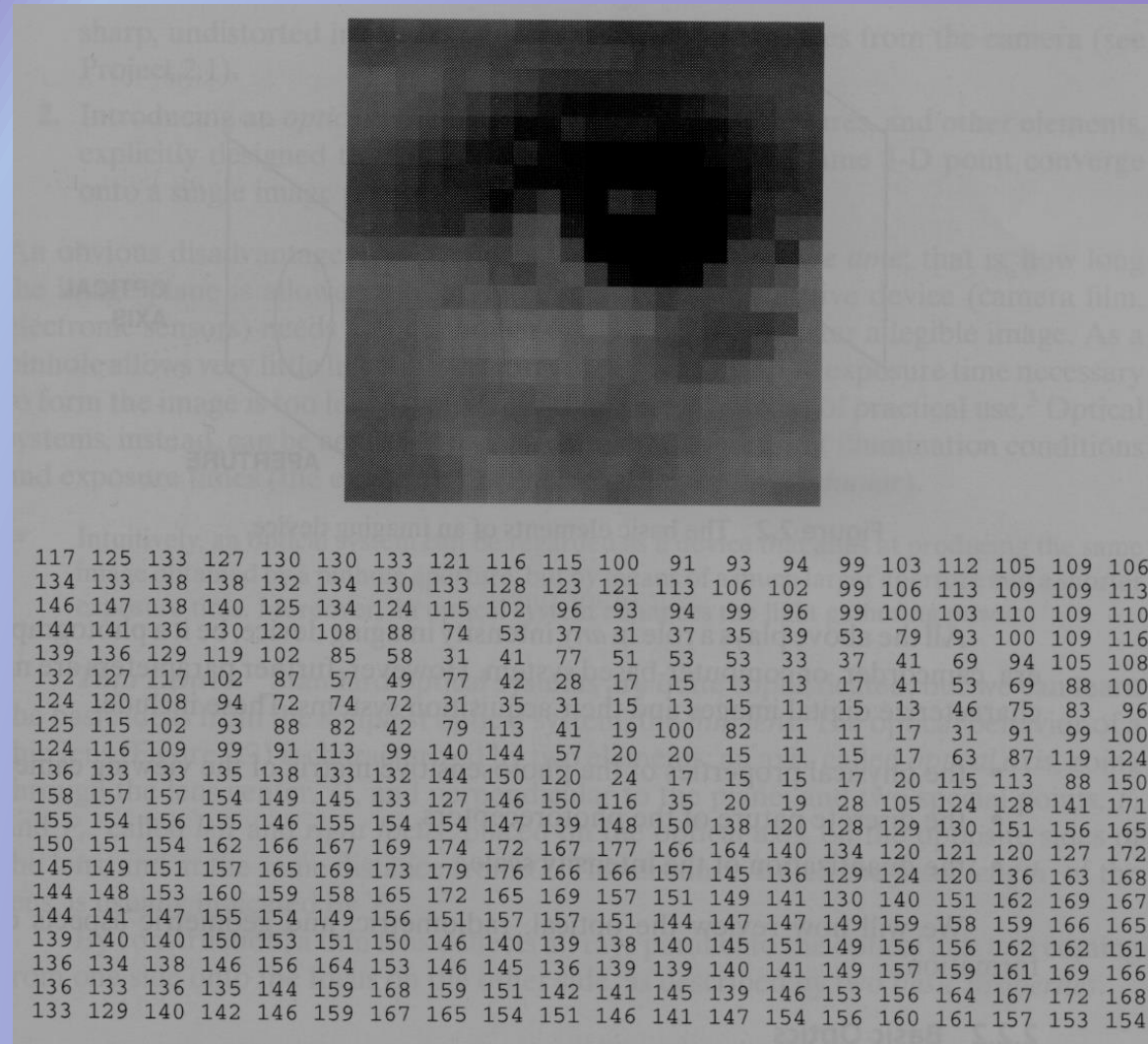


16 graylevel



32 graylevel

1 pixel = 8 bit = 256 graylevel → 0 sampai 255



- Umumnya 256 level (8 *bit/pixel*) sudah cukup untuk merepresentasikan nilai intensitas pixel.
- Jika citra digital berukuran  $N \times M$  dan setiap *pixel* kedalamannya  $b$  bit, maka kebutuhan memori untuk mereresentasikan citra adalah
$$N \times M \times b \text{ bit}$$
- Untuk citra berwarna, 256 level digunakan untuk setiap warna.
- Citra berwarna terdiri dari tiga kanal warna: – red, green, and, blue – Kombinasi ketiga warna menghasilkan persepsi warna-warna yang kita lihat.





=



Red



Green



Blue



Red

|     |     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 148 | 162 | 175 | 182 | 189 | 194 | 195 | 193 | 195 | 195 | 197 |
| 148 | 164 | 174 | 176 | 185 | 189 | 191 | 191 | 196 | 194 | 195 |
| 144 | 159 | 167 | 176 | 178 | 185 | 188 | 191 | 196 | 194 | 197 |
| 128 | 147 | 157 | 168 | 173 | 179 | 182 | 184 | 191 | 191 | 192 |
| 119 | 134 | 148 | 160 | 164 | 170 | 179 | 176 | 181 | 189 | 185 |
| 145 | 124 | 142 | 151 | 160 | 168 | 169 | 174 | 180 | 182 | 183 |
| 172 | 120 | 140 | 153 | 157 | 169 | 171 | 178 | 180 | 182 | 182 |
| 196 | 120 | 129 | 144 | 152 | 158 | 167 | 170 | 177 | 176 | 178 |
| 204 | 144 | 116 | 134 | 142 | 149 | 155 | 165 | 165 | 170 | 171 |

Green

|     |     |    |    |    |    |    |    |    |    |    |
|-----|-----|----|----|----|----|----|----|----|----|----|
| 42  | 43  | 48 | 50 | 53 | 56 | 56 | 53 | 54 | 54 | 54 |
| 50  | 49  | 51 | 47 | 53 | 55 | 56 | 55 | 59 | 55 | 54 |
| 51  | 48  | 47 | 49 | 49 | 51 | 50 | 52 | 54 | 51 | 54 |
| 53  | 48  | 45 | 49 | 50 | 52 | 50 | 48 | 51 | 50 | 50 |
| 59  | 43  | 43 | 48 | 47 | 48 | 54 | 47 | 49 | 55 | 50 |
| 100 | 42  | 41 | 42 | 44 | 46 | 45 | 46 | 50 | 52 | 50 |
| 142 | 47  | 43 | 42 | 39 | 46 | 44 | 48 | 49 | 51 | 49 |
| 185 | 65  | 44 | 42 | 42 | 43 | 48 | 46 | 50 | 48 | 49 |
| 209 | 106 | 44 | 42 | 41 | 42 | 44 | 50 | 48 | 50 | 49 |

Blue

|     |    |    |    |    |    |    |    |    |    |    |
|-----|----|----|----|----|----|----|----|----|----|----|
| 16  | 24 | 32 | 35 | 37 | 40 | 40 | 37 | 37 | 38 | 36 |
| 19  | 25 | 31 | 28 | 34 | 37 | 38 | 37 | 40 | 35 | 33 |
| 17  | 23 | 27 | 33 | 32 | 35 | 33 | 36 | 39 | 35 | 37 |
| 20  | 19 | 23 | 31 | 33 | 34 | 34 | 32 | 36 | 35 | 35 |
| 29  | 16 | 24 | 33 | 32 | 34 | 39 | 30 | 31 | 38 | 34 |
| 71  | 11 | 18 | 24 | 30 | 33 | 30 | 30 | 34 | 36 | 34 |
| 113 | 14 | 16 | 21 | 24 | 32 | 30 | 32 | 33 | 35 | 33 |
| 156 | 32 | 13 | 20 | 25 | 28 | 33 | 31 | 35 | 33 | 32 |
| 177 | 72 | 9  | 16 | 22 | 26 | 30 | 35 | 32 | 33 | 32 |



## Kode program Matlab untuk dekomposisi R, G, B dari citra berwarna

```
function fig = imColorSep(A)
% IMCOLORSEP Displays the RGB decomposition of a full-color image
%
% Syntax: fig = imColorSep(A);
%
% Example: A = imread('peppers.png');
% fig = imColorSep(A);
%
% Written by: Rick Rosson, 2007 December 26
%
% Revised: Rick Rosson, 2008 January 2
%
% Copyright (c) 2007-08 Richard D. Rosson. All rights reserved.
%
% Number of gray scale values:
N = 256;
% Make sure data type of image array is 'uint8':
A = im2uint8(A);
% Create figure window:
fig = figure;
% Display full color image:
subplot(2,2,1);
imshow(A);
title('Full Color');
```

```
% Cell array of color names:
ColorList = { 'Red' 'Green' 'Blue' };

% Gray-scale column vector: % range [ 0 .. 1 ]
gr = 0:1/(N-1):1; % increment 1/(N-1)

% Display each of the three color components:
for k = 1:3
    % color map:
    cMap = zeros(N,3);
    cMap(:,k) = gr;

    % Display monochromatic image:
    subplot(2,2,k+1);
    imshow(ind2rgb(A(:, :, k), cMap));
    title(ColorList{k});
end
end
```

