

# **Pengolahan Citra Dan Visi Komputer PCVK**

## **RTI235007**

**Pengantar Citra Digital: Representasi, Karakteristik, dan Aplikasi  
dalam Visi Komputer**

**TEACHING TEAM**  
**MATA KULIAH PENGOLAHAN CITRA DAN VISI KOMPUTER**

# **Operasi Dasar Pengolahan Citra**

- Transformasi titik: brightness, contrast, inverse, logarithmic, grayscale
- Operasi aritmatika dan logika pada citra
- Gamma correction
- Image depth dan PSNR
- Denoising dan masking

# Linier Brightness

- Operasi ini melakukan penambahan nilai pixel jika nilai brightness dinaikkan, dan mengurangkan nilai pixel jika nilai brightness diturunkan.

$$g(x,y) = f(x,y) + b$$

- $g(x,y)$  adalah nilai pixel setelah transformasi
- $f(x,y)$  adalah nilai pixel asli
- $b$  adalah nilai brightness

# Linier Brightness



# Contrast

- Kontras adalah tingkat penyebaran pixel – pixel ke dalam intensitas warna.

$$g(x,y) = a*f(x,y)+b$$

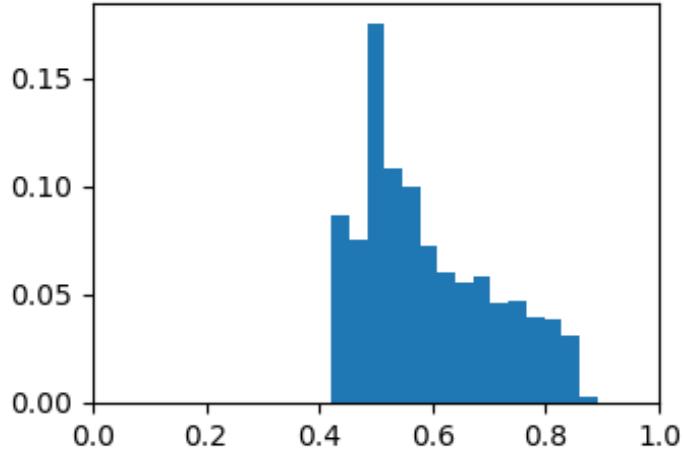
- $g(x,y)$  adalah nilai pixel setelah transformasi
- $f(x,y)$  adalah nilai pixel asli
- $a$  adalah nilai contrast
- $b$  adalah nilai brightness

# Contrast

Low contrast orginal



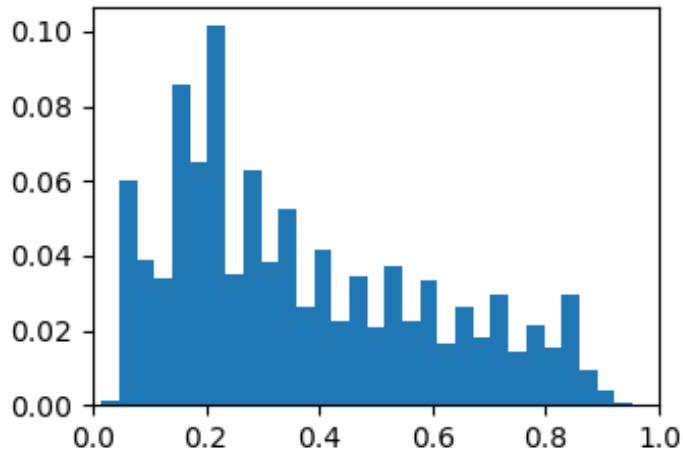
Histogram of low contrast image



Contrast Stretched



Histogram of contrast stretched image



# Logarithmic Brightness

- Transformasi Log memetakan suatu citra dengan range warna sempit menjadi lebih lebar pada citra output.

$$s = c * \log(1+r)$$

- $s$  : nilai grey-level citra output
- $c$  : konstanta
- $r$  : nilai grey-level citra input

# Inverse Color

- Proses yang digunakan untuk membalik nilai citra dengan cermin nilai pixel tengah

$$g(x) = 255 - f(x)$$

- $g(x)$  adalah citra negative
- $f(x)$  adalah citra asli

# Inverse Color



# Grayscale

- Grayscale adalah citra keabuan dengan nilai intensitas kedalaman pixel 8 bit

$$Grayscale_{avg} = \frac{(R + G + B)}{3}$$

$$Grayscale_{Lightness} = \frac{\max[R, G, B] + \min[R, G, B]}{2}$$

$$Grayscale_{Luminance} = 0.21R + 0.72G + 0.07B$$

# Fungsi Transformasi Intensitas Dasar

- **Invers Citra**
  - $S = L-1-r$ , dengan nilai intensitas  $[0, L-1]$
- **Transformasi Log**
  - $S = c \log(1+r)$ ,  $c$  adalah konstanta dengan asumsi  $r \geq 0$
- **Transformasi Gamma (Power-Low)/Gamma Correction**
  - $S = cr^\gamma$
  - $S = 255 c \left(\frac{r}{255}\right)^{1/\gamma}$

# Invers Citra

Northpole Image



(a)

Negative of Northpole

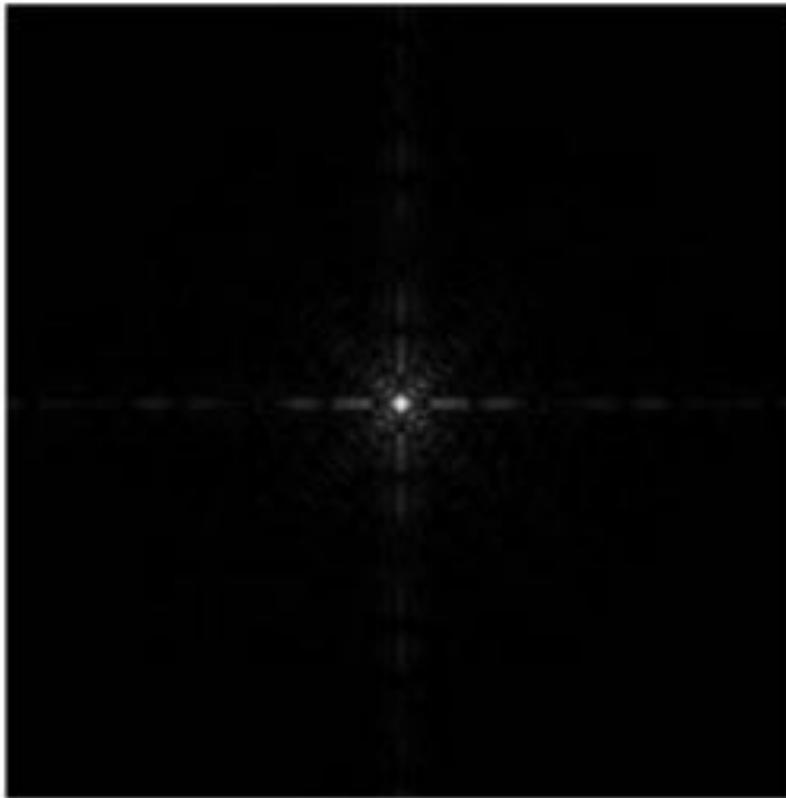


(b)

(a) Gambar Northpole

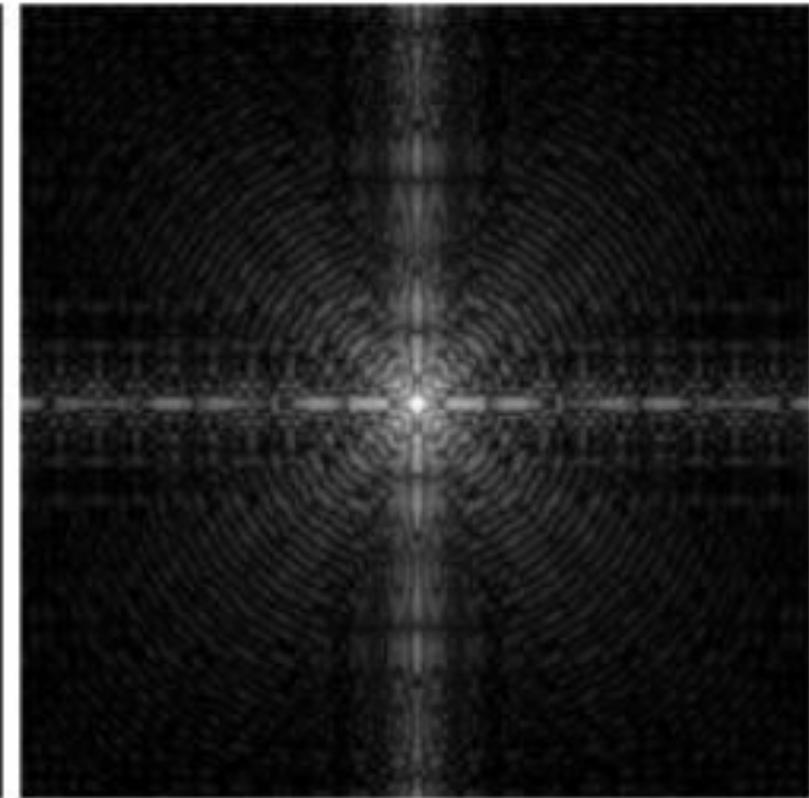
(b) Gambar Negative of Northpole yang didapatkan dengan Invers Citra

# Transformasi Log



(a)

(a) Fourier spectrum



(b)

(b) Hasil dari menerapkan log transformasi dengan  $c = 1$

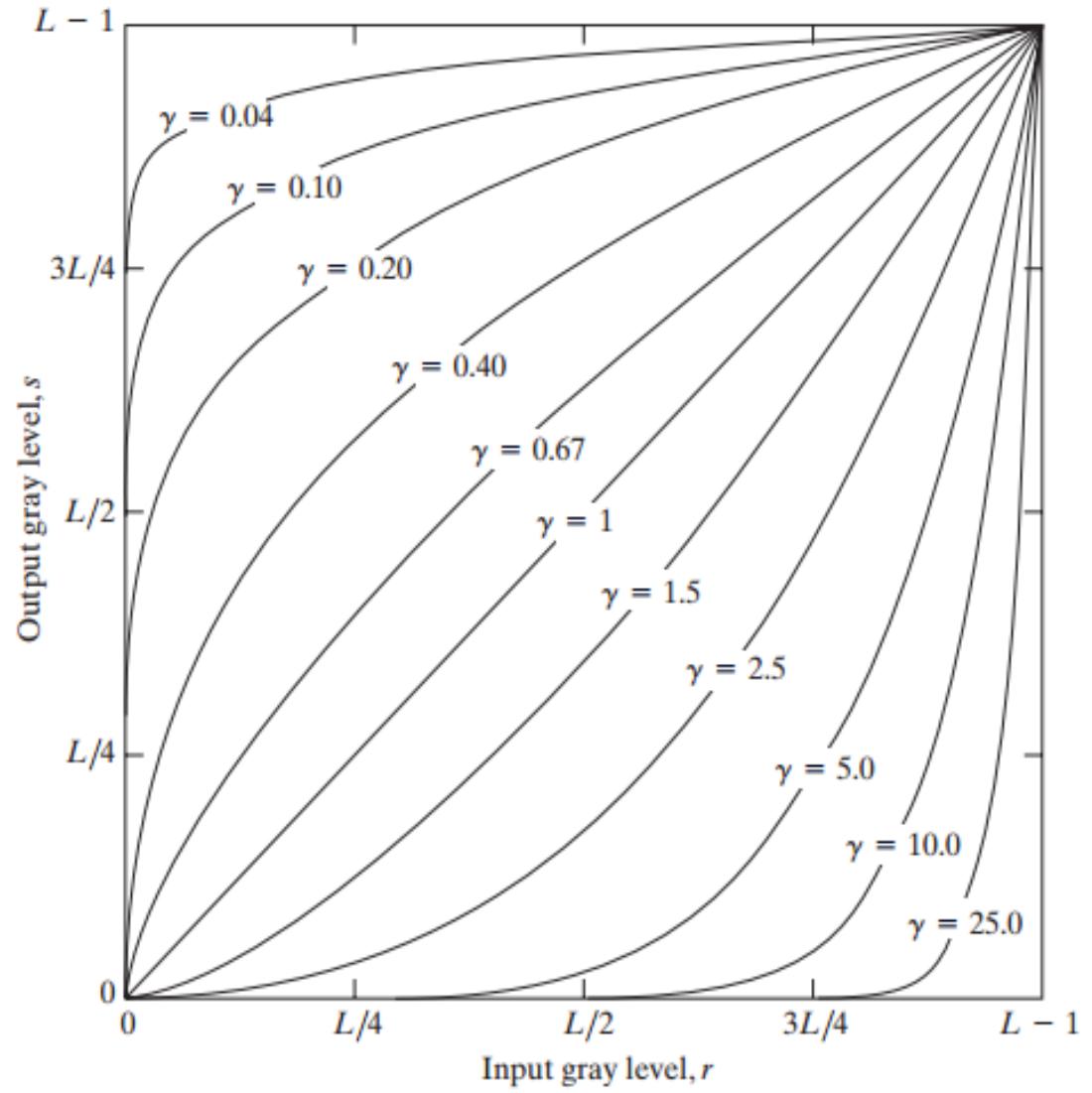
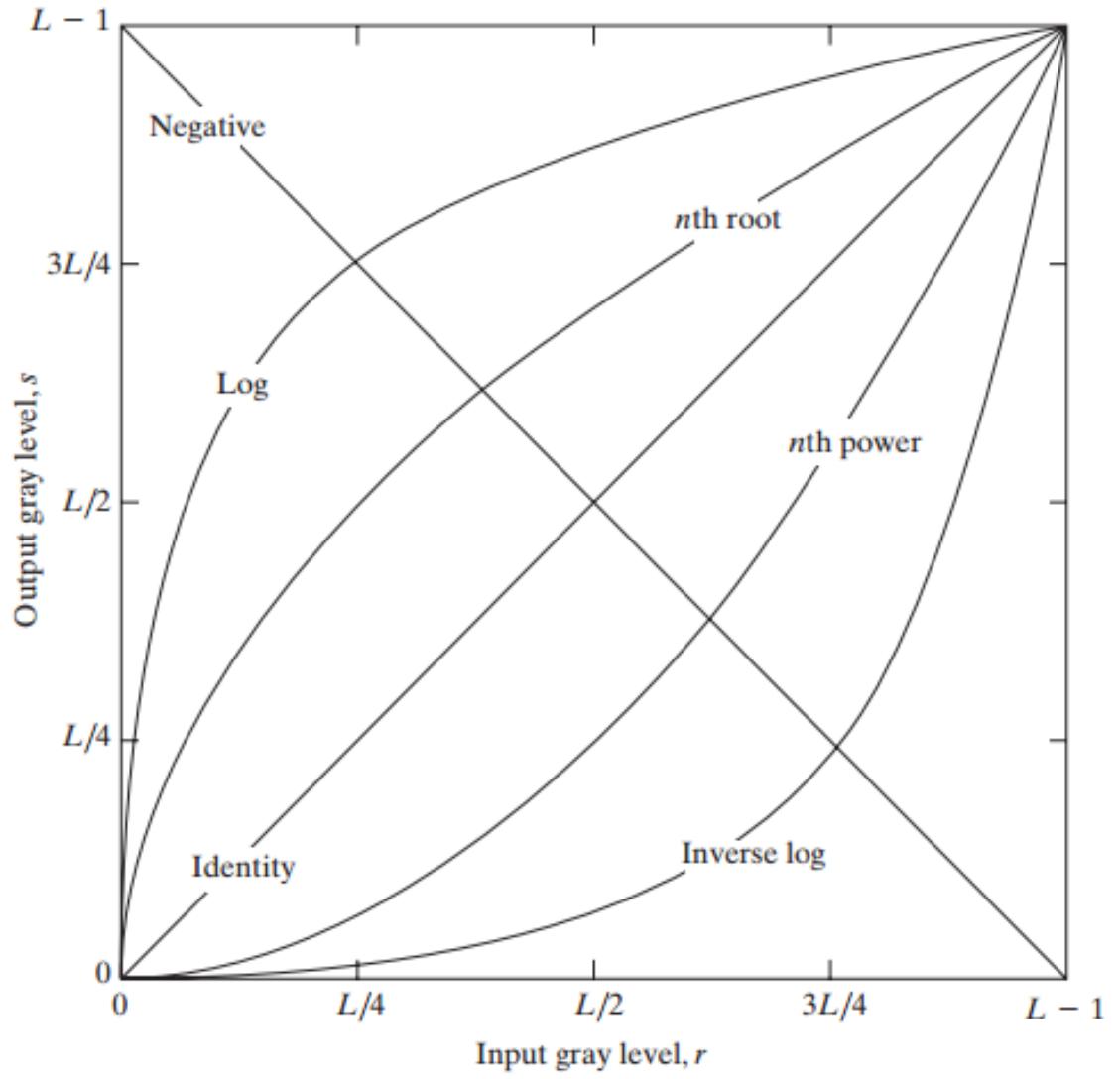
# Gamma Correction

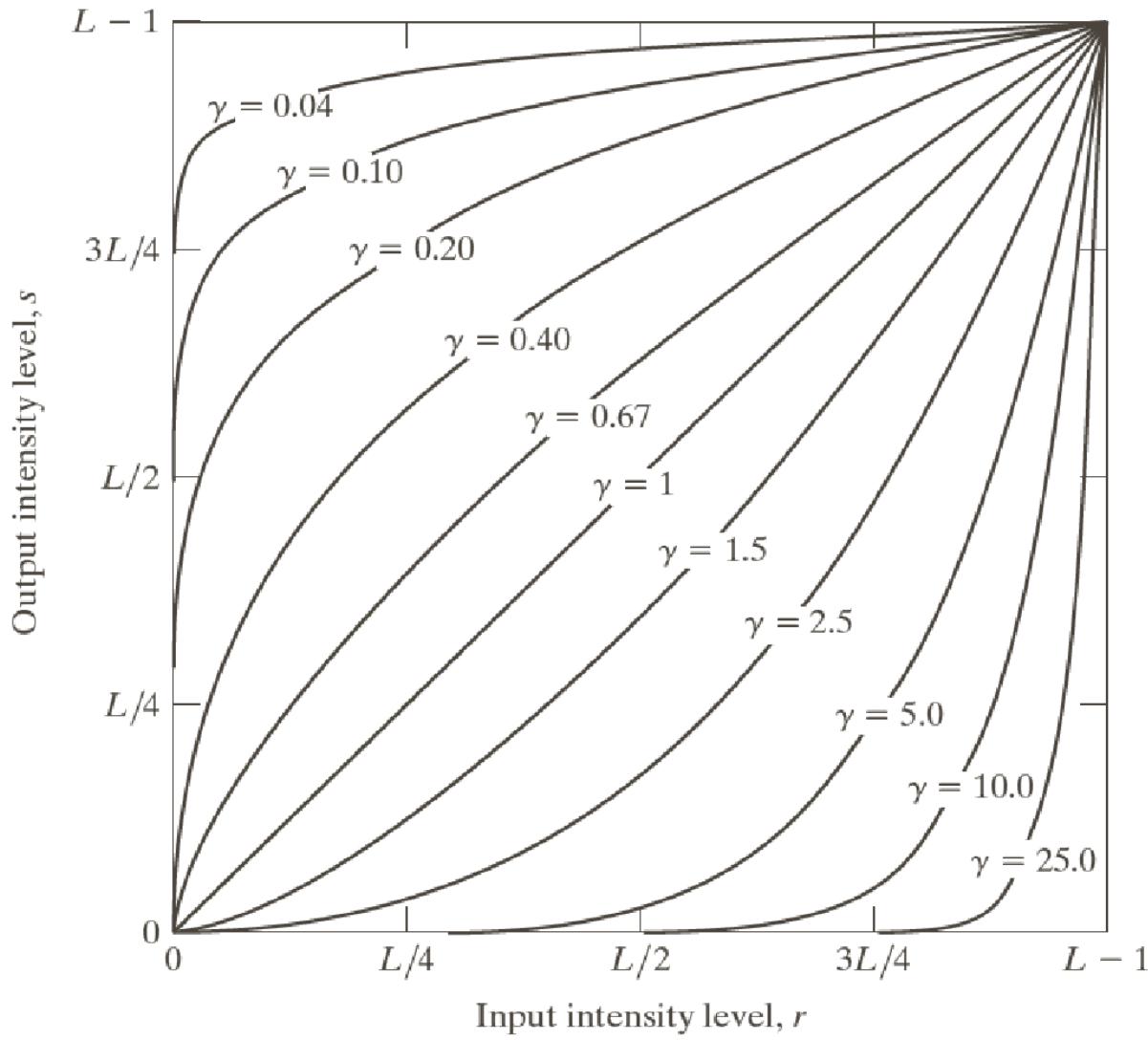
- Gamma correction adalah salah satu teknik dalam operasi citra digital yang digunakan untuk memperbaiki dan mengoreksi tingkat kecerahan dan kontras pada sebuah gambar.
- Teknik ini penting karena monitor dan kamera cenderung tidak selalu menghasilkan tingkat kecerahan yang linier.
- Gamma Correction melibatkan penggunaan fungsi matematis untuk mengubah nilai piksel dalam gambar, mengkompensasi perbedaan antara kecerahan sebenarnya dan kecerahan yang diharapkan dalam sebuah gambar.
- Gamma Correction banyak digunakan dalam pengolahan citra digital dan juga dalam teknologi tampilan seperti monitor. Digunakan dalam berbagai konteks: perbaikan citra gelap, peningkatan kualitas gambar, dan penyesuaian tampilan.

# Gamma Correction

- Gamma (Power-Law Transformation), disimbolkan dengan huruf yunani  $\gamma$ , dijelaskan sebagai hubungan antara masukan dan keluaran yang dihasilkan.
- Masukan yang dimaksud adalah nilai intensitas RGB dari citra. Hubungan antara masukan dan keluaran yang dimaksud adalah keluarannya proporsional dengan masukan yang dipangkatkan dengan nilai gamma.
- Nilai gamma yang umum digunakan berkisar antara 1/2 hingga 2. Nilai gamma yang kurang dari 1 akan menghasilkan kontrast rendah, sementara nilai gamma yang lebih dari 1 akan menghasilkan kontrast yang tinggi.
- Rumus dari Gamma adalah sebagai berikut:

$$l' = 255x \left( \frac{l}{255} \right)^{1/\gamma}$$



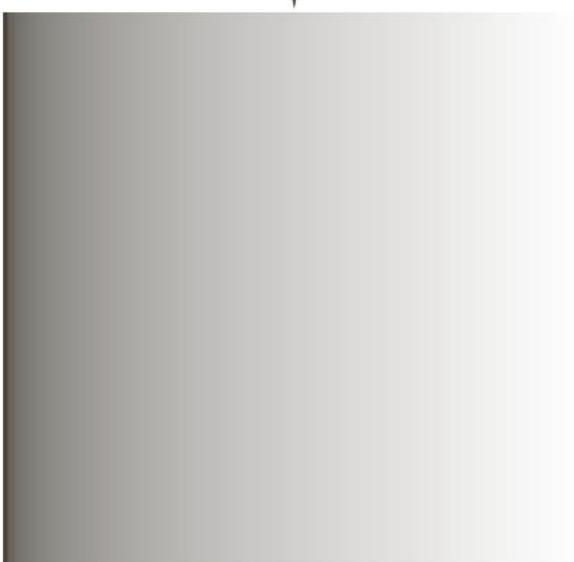


**FIGURE 3.6** Plots of the equation  $s = cr^\gamma$  for various values of  $\gamma$  ( $c = 1$  in all cases). All curves were scaled to fit in the range shown.



Original image

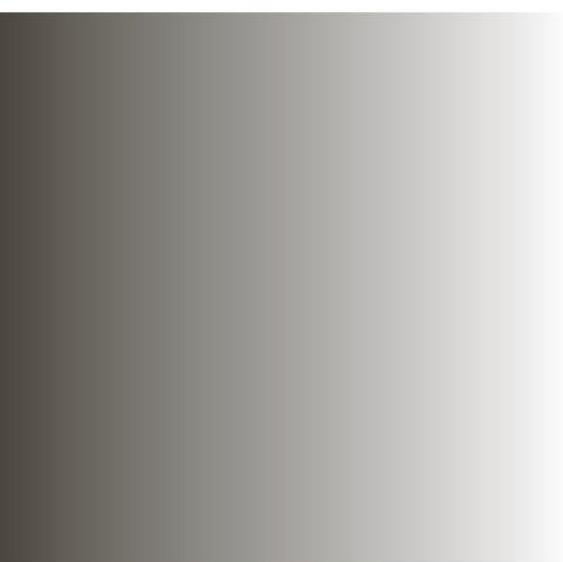
↓  
Gamma  
correction



Gamma-corrected image



Original image as viewed  
on monitor

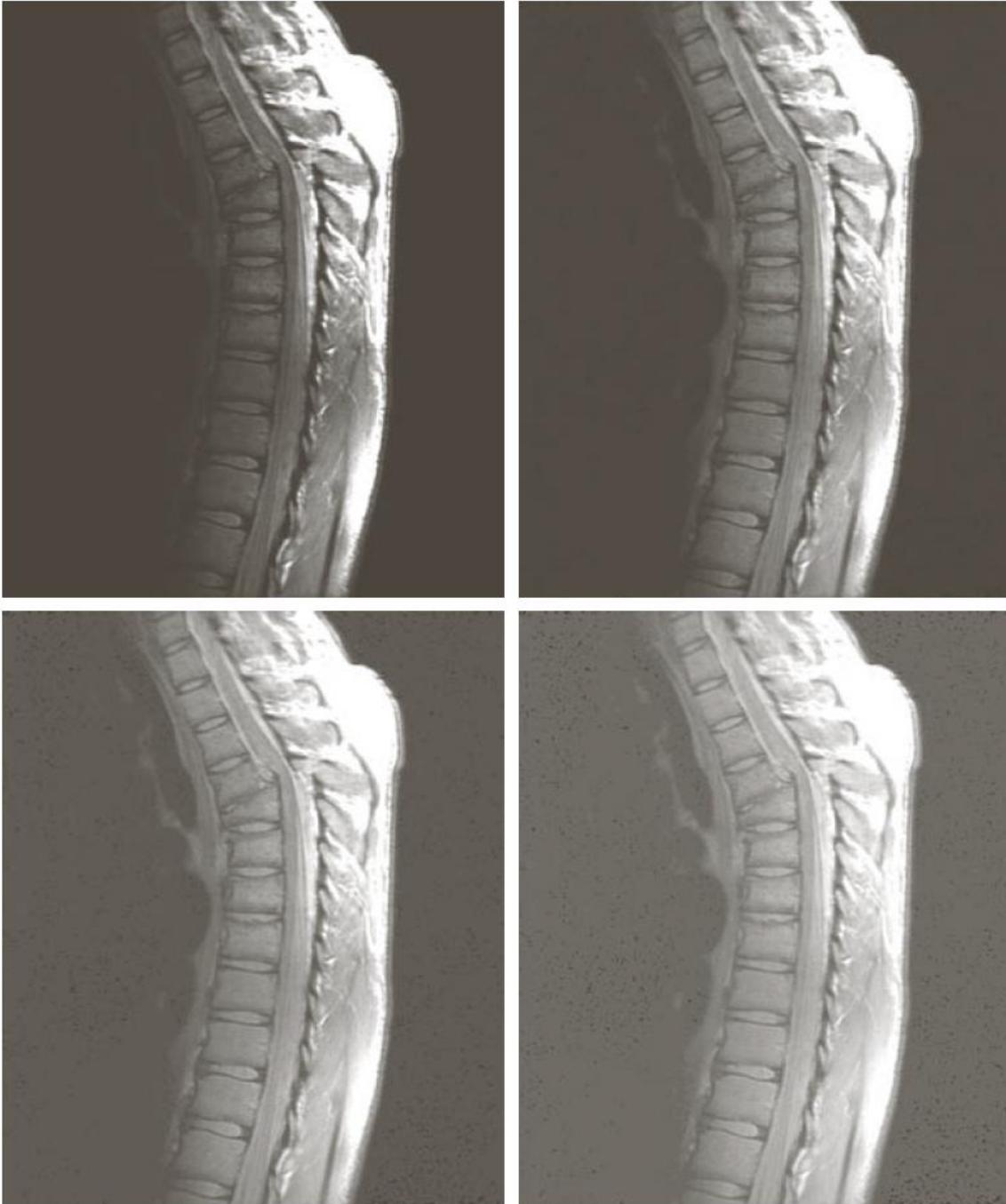


Gamma-corrected image as  
viewed on the same monitor

a	b
c	d

**FIGURE 3.7**

(a) Intensity ramp image. (b) Image as viewed on a simulated monitor with a gamma of 2.5. (c) Gamma-corrected image. (d) Corrected image as viewed on the same monitor. Compare (d) and (a).



a b  
c d

**FIGURE 3.8**  
(a) Magnetic resonance image (MRI) of a fractured human spine.  
(b)–(d) Results of applying the transformation in Eq. (3.2-3) with  $c = 1$  and  $\gamma = 0.6, 0.4$ , and  $0.3$ , respectively.  
(Original image courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

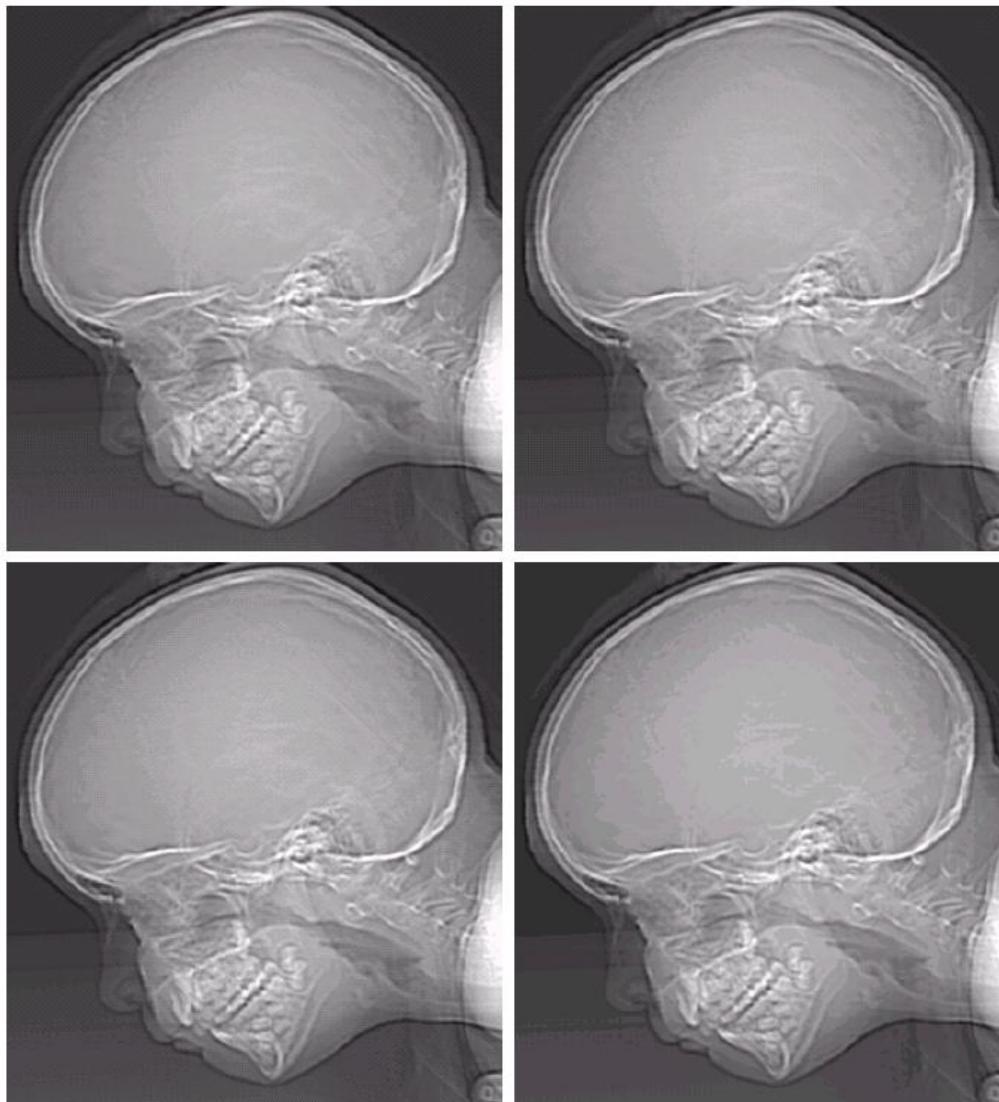


a b  
c d

**FIGURE 3.9**  
(a) Aerial image.  
(b)–(d) Results of  
applying the  
transformation in  
Eq. (3.2-3) with  
 $c = 1$  and  
 $\gamma = 3.0, 4.0$ , and  
 $5.0$ , respectively.  
(Original image  
for this example  
courtesy of  
NASA.)

# Image Bit-Depth

- Bit-Depth terkait dengan kuantisasi jumlah bit pada citra.
- Jika menggunakan 8-bit kuantisasi, maka jumlah warna yang mungkin adalah  $2^8=256$  warna.
- Bit-Depth memengaruhi sejauh mana gambar dapat mereproduksi detail dan tingkat warna yang berbeda.
- Semakin tinggi bit-depth, semakin banyak warna yang dapat direpresentasikan dalam gambar, dan semakin tinggi tingkat detail yang dapat dihasilkan.



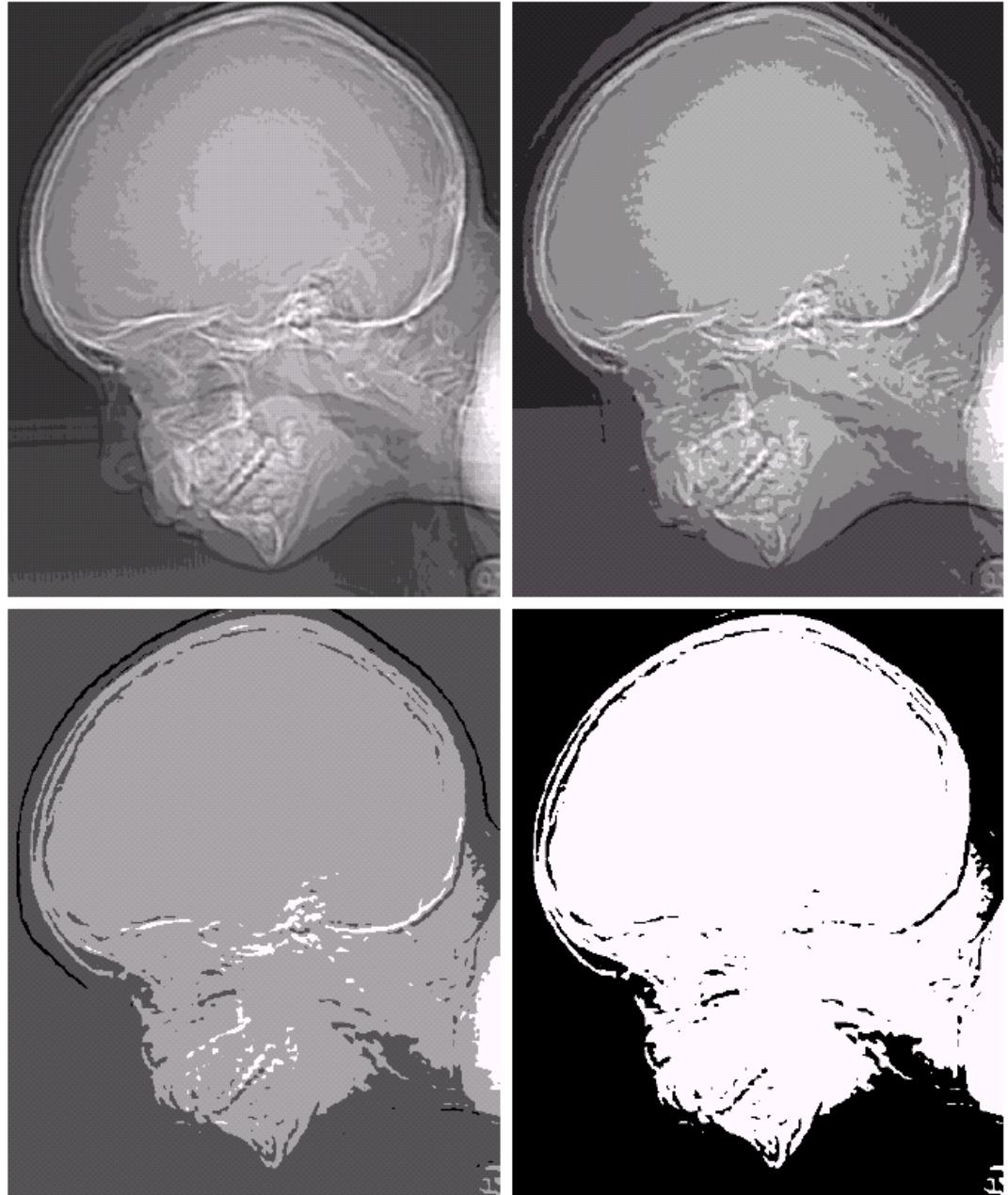
a b  
c d

**FIGURE 2.21**  
(a)  $452 \times 374$ ,  
256-level image.  
(b)–(d) Image  
displayed in 128,  
64, and 32 gray  
levels, while  
keeping the  
spatial resolution  
constant.

# Image Bit-Depth

e f  
g h

**FIGURE 2.21**  
*(Continued)*  
(e)–(h) Image displayed in 16, 8, 4, and 2 gray levels. (Original courtesy of Dr. David R. Pickens, Department of Radiology & Radiological Sciences, Vanderbilt University Medical Center.)



Perhatikan image disamping. Jika terkuantisasi 1bit, maka jumlah warna yang mungkin hanya 2 yaitu hitam dan putih. Jika 2bit, maka jumlah warna yang mungkin adalah 4 (hitam, abu-abu muda, abu-abu tua, dan putih).

# Image Bit-Depth

- Berikut adalah rumus yang digunakan:

$$level = \frac{255}{2^{bit\_depth} - 1}$$

$$C' = round \left( \left( \frac{C}{level} \right) * evel \right)$$

# Image Bit-Depth

Bit Depth (n)	Varian (biner)	Varian (desimal)	Banyaknya varian ( $2^n$ )
1	0,1	0,1	2
2	00,01,10,11	0,1,2,3	4
3	000,001,010,011,100,101,110,111	0,1,2,3,4,5,6,7	8
4	0000,0001,0010,0011,0100...,1111	0,1,2,3,4,5,6,7,...,15	16
5	00000,00001,00010,...,11111	0,1,2,3,4,5,6,7,...,31	32
6	000000,000001,000010,...,111111	0,1,2,3,4,5,6,7,...,63	64
7	0000000,0000001,...,1111111	0,1,2,3,4,5,6,7,...,127	128
8	00000000,00000001,...,11111111	0,1,2,3,4,5,6,7,...,255	256

# PSNR

- PSNR (Peak Signal-to-Noise Ratio) adalah ratio antara nilai power maksimum dari citra dengan power dari citra yang terkena noise yang mempengaruhi kualitas dari citra tersebut.
- Semakin tinggi nilai PSNR, semakin baik kualitas citra tersebut.
- Untuk menghitung PSNR dari citra, yang perlu dilakukan adalah membandingkan image hasil denoising / terkompress dengan citra ideal yang memiliki power maksimum (dalam hal ini adalah citra asli).
- PSNR sering digunakan dalam berbagai aplikasi seperti kompresi citra, pemrosesan gambar, dan pemrosesan video untuk mengukur tingkat degradasi kualitas yang terjadi selama proses kompresi atau pengolahan.

# PSNR

- PSNR dirumuskan sebagai berikut:

$$PSNR = 10 \log_{10} \left( \frac{L - 1}{MSE} \right) = 20 \log_{10} \left( \frac{L - 1}{RMSE} \right)$$

- L : jumlah intensitas warna yang mungkin dari citra.
- MSE : mean square error dengan rumus:

$$MSE = \frac{1}{MN} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (Original(i, j) - enoised(i, j))^2$$

- RMSE : root mean square error yaitu akar dari MSE

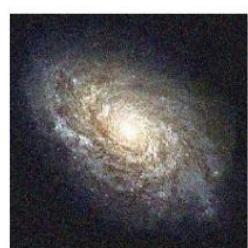
# PSNR

```
def PSNR(img1, img2):
    mse = np.mean((img1 - img2) ** 2)
    if(mse == 0): # MSE 0 maka tidak ada noise sama sekali, sehingga PSNR tidak memiliki arti
        return 100
    max_pixel = 255.0
    psnr = 20 * log10(max_pixel / sqrt(mse))
    return psnr
```

# Average Denoising

- Salah satu teknik pengolahan citra digital yang digunakan untuk mengurangi noise atau derau pada citra.
- Teknik ini berfokus pada perataan (smoothing) citra dengan cara menggantikan nilai piksel dengan rata-rata dari sejumlah nilai piksel di sekitarnya.
- Operasi ini melakukan perhitungan nilai rata-rata tiap pixel yang berkoordinat sama dalam sebuah kumpulan citra.
- Operasi ini biasanya dilakukan untuk menghilangkan noise dari faktor eksternal yang terjadi saat akuisisi citra.
- Average denoising biasa digunakan pada citra yang memiliki noise eksternal yang muncul akibat proses distribusi dari sumber ke tujuan.
- Selain itu sering digunakan dalam berbagai aplikasi, termasuk pengolahan citra medis, pemrosesan gambar, dan pemrosesan video.

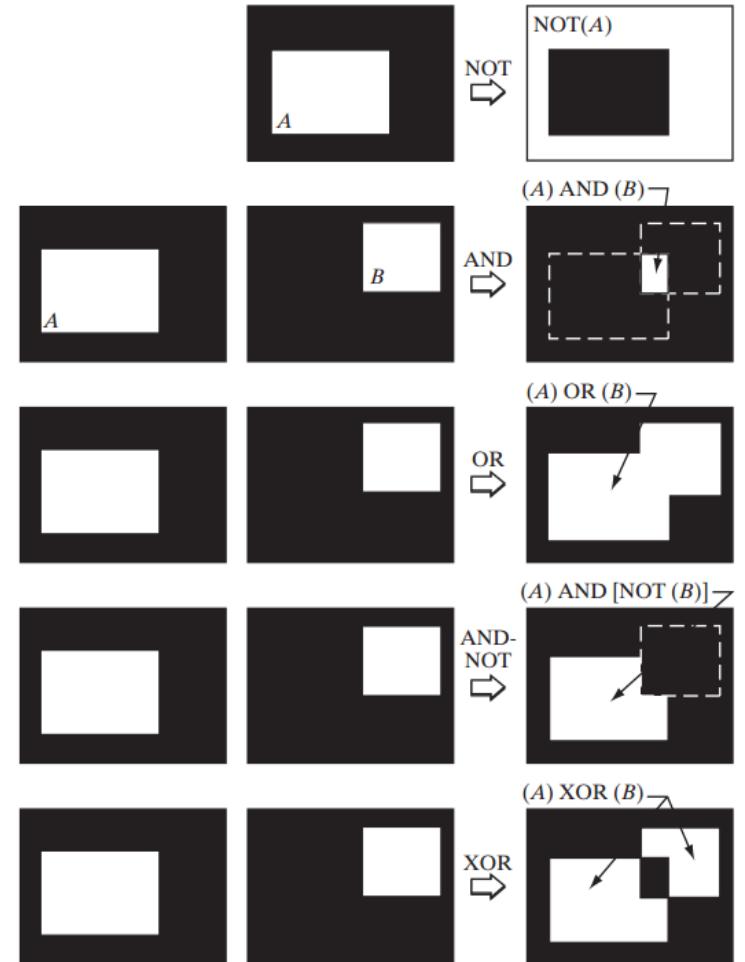
# Average Denoising



- Pada gambar disamping ditunjukkan beberapa gambar yang berisi Gaussian Noise.
- Dengan Average Denoising, akan dibuat gambar baru yang akan menghilangkan noise yang muncul dari gambar-gambar bernoise tersebut.
- Nilai PSNR dapat digunakan untuk melihat ratio power dari citra denoised dibandingkan dengan citra asli tanpa noise.

# Operasi Boolean (Image Masking)

- Image masking adalah contoh dari operasi Logika yang digunakan untuk mengolah citra.
- Beberapa operator logika yang sering digunakan pada pengolahan citra adalah operator OR, AND, dan NOT. Pada Image Masking, operator yang biasa digunakan adalah operator AND.
- Jika diasumsikan dua region (sets) A dan B terdiri dari pixel foreground, maka operasi OR dari dua set ini adalah dapat berisi nilai dari A atau B. Gambar berikut mengilustrasikan operasi-operasi Logika pada citra.
- Operasi Boolean (Image Masking) ini sering digunakan dalam berbagai aplikasi pengolahan citra seperti segmentasi objek, pencitraan medis, deteksi tepi, dan banyak lagi.

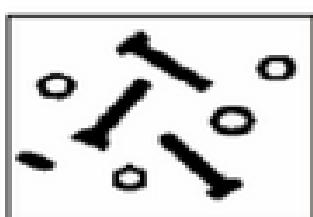
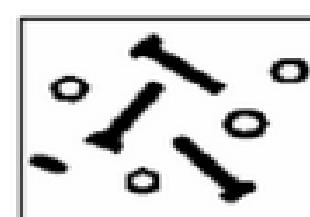
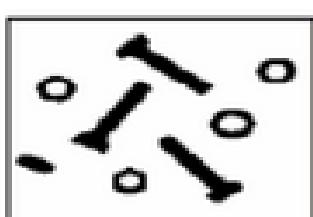
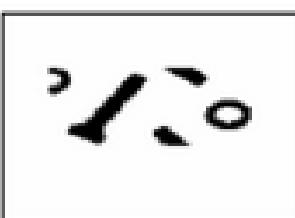
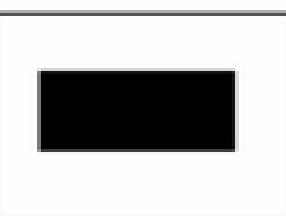
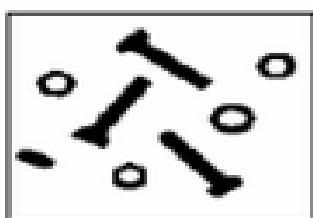
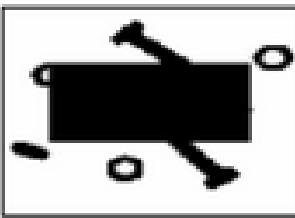
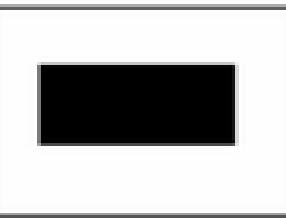
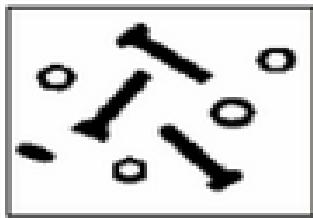


# Operasi Boolean (Image Masking)

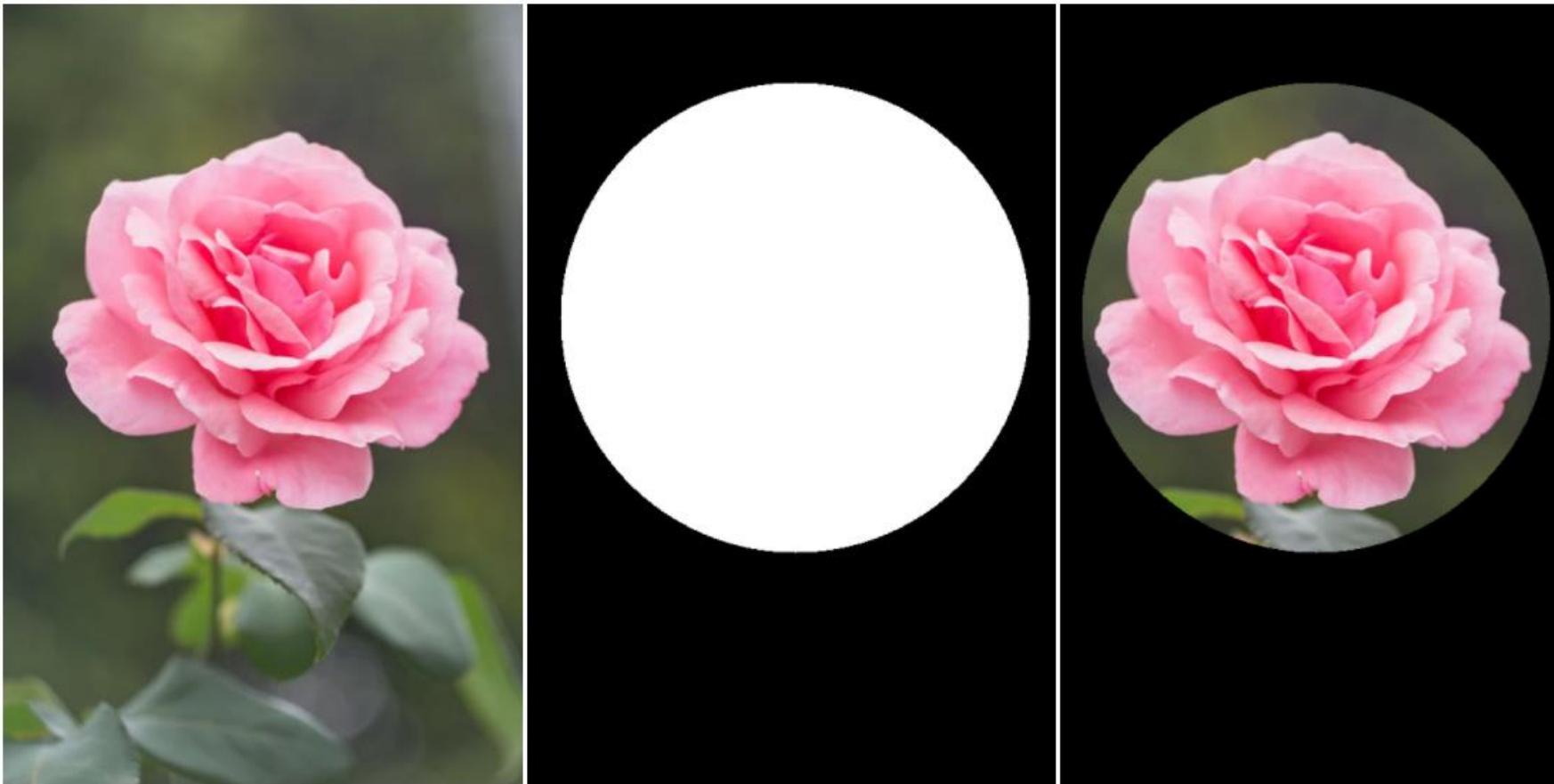
Berikut adalah penjelasan lebih rinci tentang Operasi Boolean (Image Masking) dalam operasi citra digital:

- AND (Konjungsi): Pada operasi AND, piksel pada citra hasil akan bernilai 1 (putih) hanya jika kedua piksel yang bersesuaian pada citra input dan masker memiliki nilai 1. Digunakan untuk memisahkan objek dari latar belakang.
- OR (Disjungsi): Operasi OR menghasilkan piksel dengan nilai 1 jika salah satu atau kedua piksel pada citra input dan masker memiliki nilai 1. Berguna untuk menonjolkan objek pada latar belakang yang berbeda.
- NOT (Negasi): Operasi NOT menghasilkan citra hasil dengan membalik nilai piksel pada citra input. Piksel hitam menjadi putih dan sebaliknya. Digunakan untuk mengubah kontras atau membalik citra.

# Operasi Boolean



# AND Image Masking



# REFERENSI

Digital Image Processing, Rafael C. Gonzalez and Richard E. Woods

Rayra, H., Yassir, Y., & Rachmawati, R. (2023). PENGGUNAAN KOREKSI GAMMA DENGAN METODE ROBERT, PREWITT, DAN SOBEL UNTUK PENYEMPURNAAN GAMBAR PADA CITRA DALAM AIR. Jurnal TEKTRO, 7(1), 65-71.

<https://learn.microsoft.com/id-id/windows/win32/direct3ddxgi/using-gamma-correction>

<http://preservationtutorial.library.cornell.edu/tutorial/intro/intro-04.html>

<https://etc.usf.edu/techease/win/images/what-is-bit-depth/>

<https://www.geeksforgeeks.org/denoising-techniques-in-digital-image-processing-using-matlab/>

<https://www.olympus-lifescience.com/en/microscope-resource/primer/java/digitalimaging/processing/imageaveraging/>