



# TRANSFORMASI DAN MODEL WARNA CITRA DIGITAL

COMPUTER VISION



# REPRESENTASI BIT CITRA

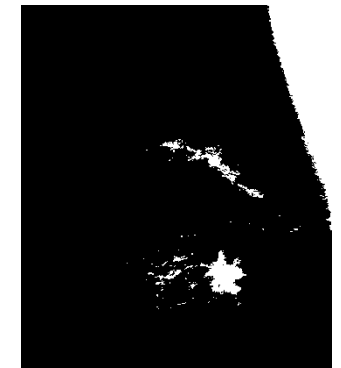
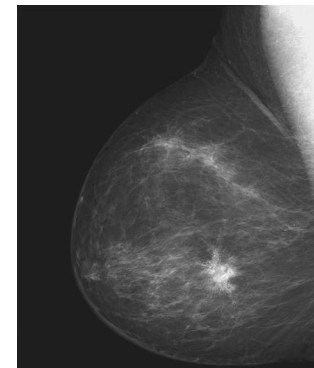
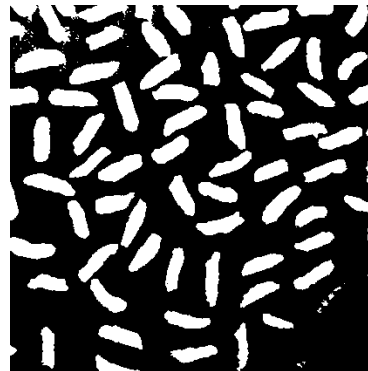
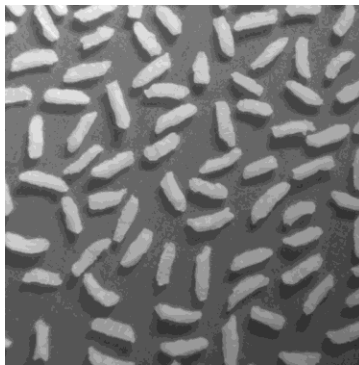
Bit	Range	Keterangan
1	0 - 1	Citra Biner
8	0 - 255	Citra abu-abu (grayscale)
12	0 - 4096	High quality grayscale
16	0 - 65535	Very high quality grayscale
32	(0.0 - 1.0)	Floating point format
8+8+8	3 x 1-255	"24 bit True Color" (monitor)



Citra abu-abu



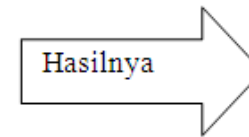
Citra biner



# CITRA NEGATIF

7	5	4	7	2
1	7	7	6	3
4	0	3	7	4
7	3	5	4	3
5	4	5	1	4

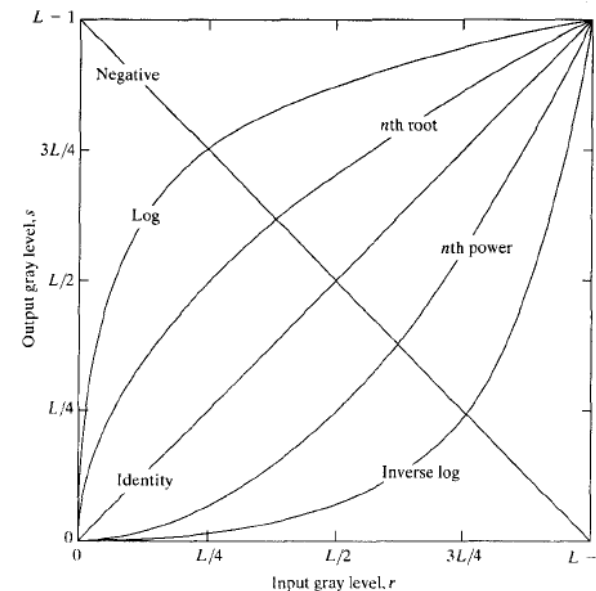
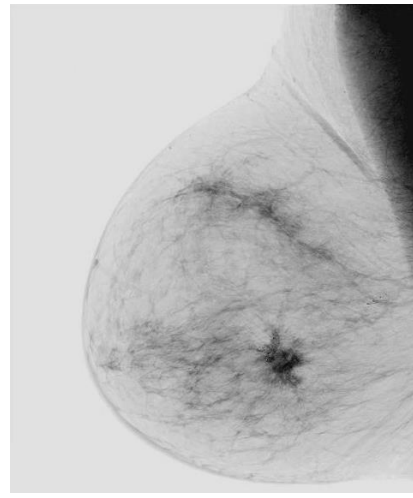
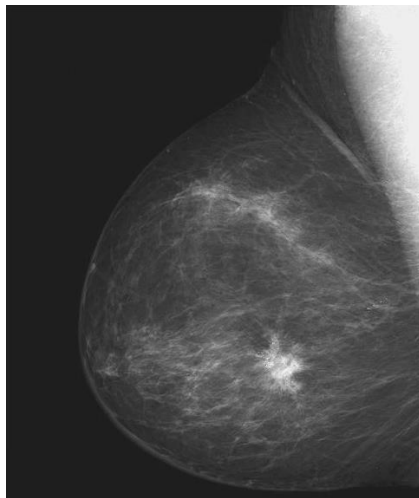
Citra asli



0	2	3	0	5
6	0	0	1	4
3	7	4	0	3
0	4	2	3	4
2	3	2	6	3

Citra negatif

- Seperti halnya film negatif.
- Hasil pengambilan gambar dengan kamera konvensional yang membalik citra asli.
- Jika terdapat citra dengan jumlah gray level  $L$  dengan range  $[0, L-1]$  maka citra negatif didapat dari transformasi negatif seperti pada gambar (garis diagonal) dengan persamaan:  $s = L - 1 - r$



Untuk  $L = 256$ , maka citra negatif didapatkan:  
 $\gg j = 256 - 1 - f;$

# TRANSFORMASI LOG

- Sangat berguna dalam penggambaran grafik ketika pada deretan nilai, di samping ada selisih nilai yang sangat kecil, juga ada selisih nilai yang sangat besar,
  - Sehingga ketika digambar dalam grafik maka selisih yang kecil akan sulit untuk dilihat.
- Kurva log yang terdapat pada gambar.
- Nilai input gray level yang tinggi akan menyesuaikan dengan output gray levelnya sesuai grafik.
- Meningkatkan nilai citra yang gelap dan mengompres nilai citra yang sangat tinggi.
- Transformasi Inverse log merupakan kebalikan dari transformasi log
- Persamaan dari transformasi log adalah:  $s = c \log(I + r)$

80	255	4	134	23
10	70	170	180	200
45	15	31	167	68
7	3	231	90	100
97	49	8	250	123

Citra asli

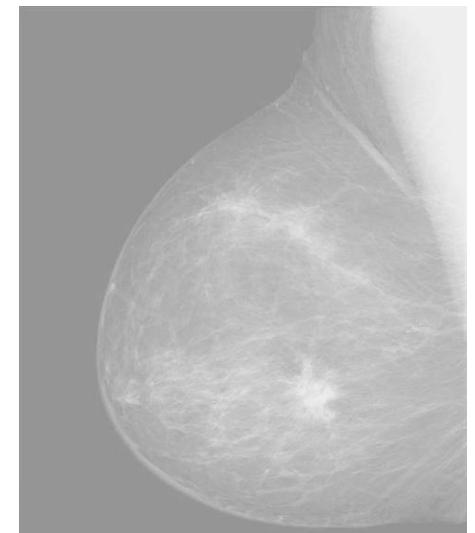
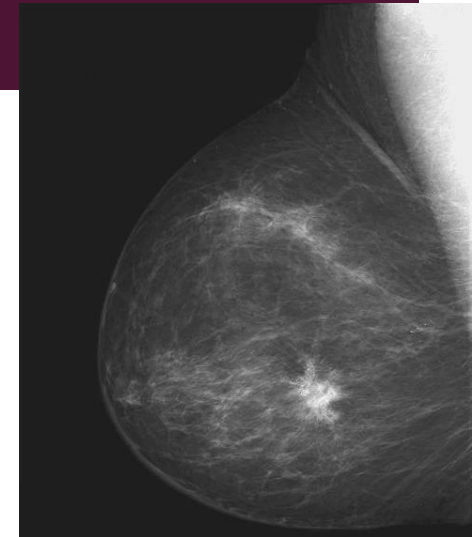
Hasilnya

190	240	70	210	140
100	190	220	230	230
170	120	150	220	180
90	60	240	200	200
200	170	100	240	210

Citra setelah dilakukan transformasi log

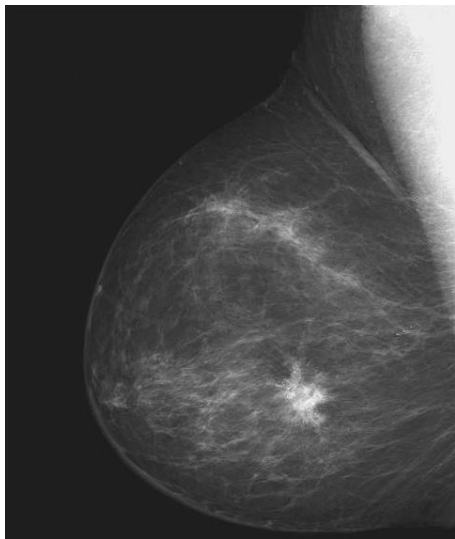
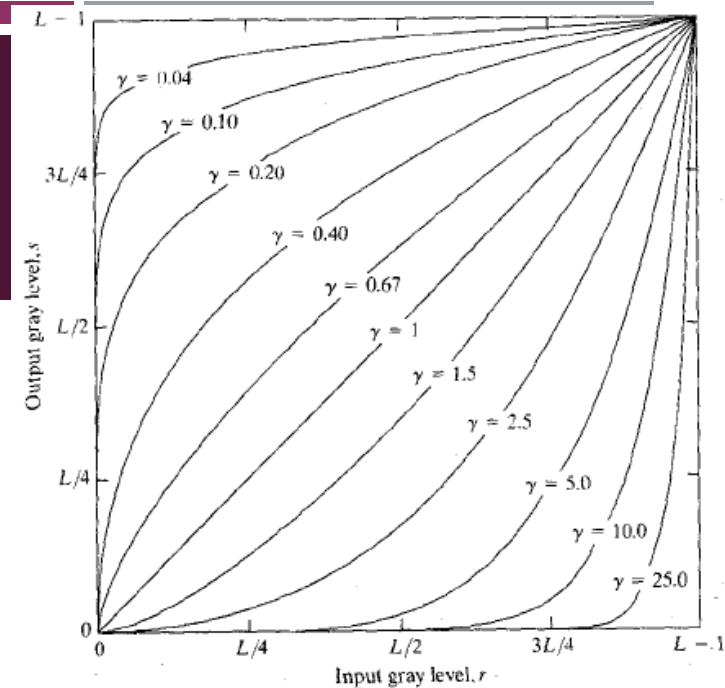
Gambar 3.4 Transformasi log pada citra dengan gray level  $L = 256$  dan  $c = 100$

```
>> g=100*log10(1+f);
```

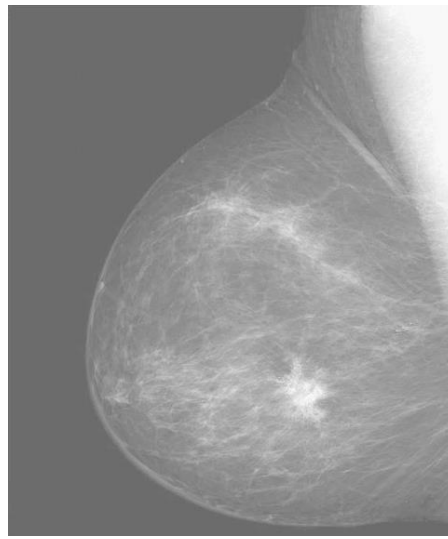


# TRANSFORMASI POWER-LAW

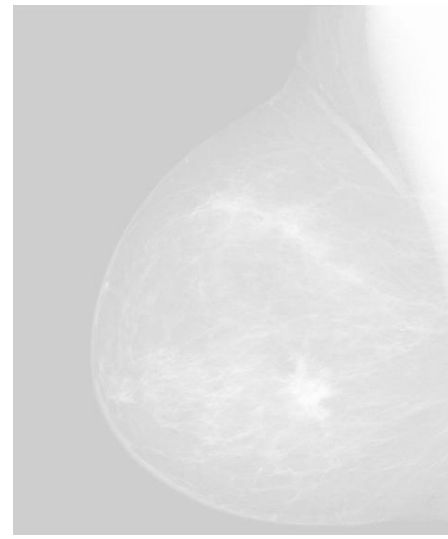
- Bentuk dasar:  $s = cr^\gamma$ 
  - $c$  dan  $\gamma$  merupakan konstanta positif.
- Memetakan range sempit dari input gray level menjadi range yang lebar pada output gray level-nya.
- Mirip dengan transformasi log, tapi dengan  $\gamma$  transformasi power-law dapat mempunyai variasi kurva yang lebih banyak daripada transformasi log.



gamma = 1



gamma = 0.4



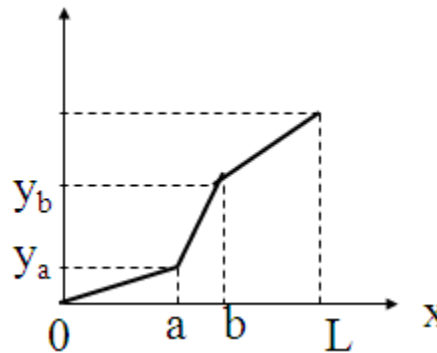
gamma = 0.1

```
>> g = im2double(f);  
>> g2= 1 * g.^0.4;
```

# CONTRAST STRETCHING

- Berguna untuk meningkatkan *dynamic range* dari citra yang telah diproses.
- Tidak seperti transformasi sebelumnya yang memproses semua input gray level, pada contrast stretching yang diproses bisa sebagian dari input gray level sesuai dengan grafik yang digunakan

$$y = \begin{cases} \alpha x & 0 \leq x < a \\ \beta(x - a) + y_a & a \leq x < b \\ \gamma(x - b) + y_b & b \leq x < L \end{cases}$$



```
>> g=f;
>> g(g<50)=g(g<50)*0.2;
>> g(g>=50 & g<150)=(g(g>=50 & g<150)-
50)*2+30;
>> g(g>=150)=(g(g>=150)-150)*1+200;
```

$a = 50, b = 150, \alpha = 0.2, \beta = 2, \gamma = 1, y_a = 30, y_b = 200$



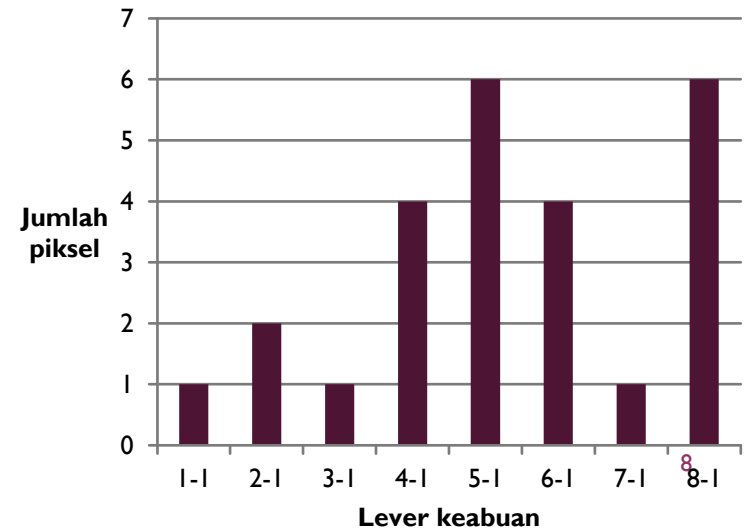
# HISTOGRAM

- Histogram adalah dasar dari sejumlah teknik pemrosesan citra pada domain spasial, seperti perbaikan, kompresi dan segmentasi citra.
- Histogram dari suatu citra digital dengan range tingkat  $[0 \dots L-1]$  adalah sebuah fungsi diskrit  $h(r_k) = n_k$ 
  - $r_k$  adalah tingkat keabuan ke- $k$
  - $n_k$  adalah jumlah piksel dalam citra yang memiliki tingkat keabuan  $r_k$ .
- Normalisasi histogram dilakukan dengan membagi setiap nilai  $n_k$  dengan total jumlah piksel dalam citra, yang dinyatakan dengan  $n$ .
  - Histogram yang sudah dinormalisasi dinyatakan dengan  $p(r_k) = n_k/n$ , untuk  $k=0, 1, \dots, L-1$ .
- $p(r_k)$  menyatakan estimasi probabilitas kemunculan tingkat keabuan  $r_k$ .
- Jumlah dari semua komponen “normalized histogram” sama dengan 1.

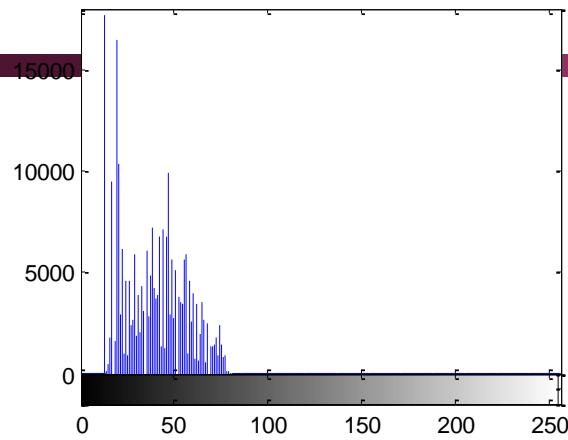
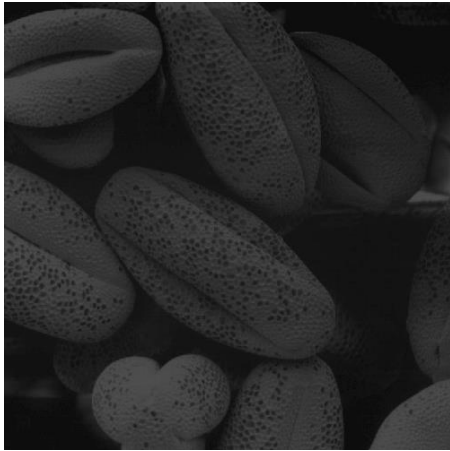
# HISTOGRAM

- Sumbu horisontal dari histogram menyatakan nilai tingkat keabuan  $r_k$ .
- Sumbu vertikal menyatakan nilai dari  $h(r_k)=n_k$  atau  $p(r_k) = n_k/n$  (jika nilainya dinormalisasi).

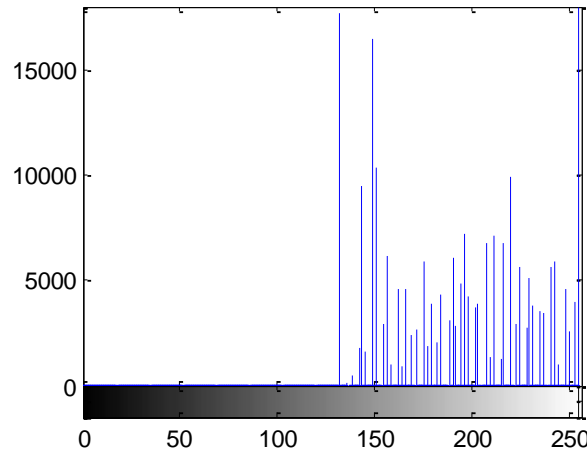
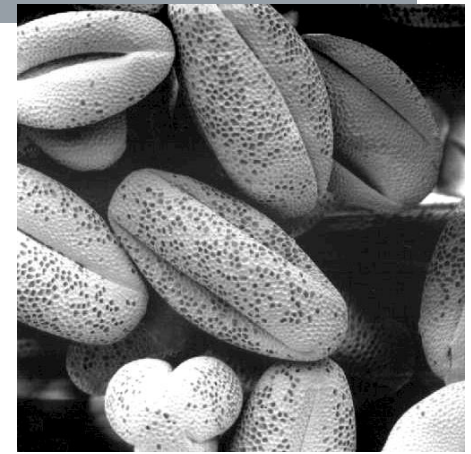
					gray	jum. pik	norm
7	5	4	7	2	$h(0)$	1	0.04
1	7	7	6	3	$h(1)$	2	0.08
4	0	3	7	4	$h(2)$	1	0.04
7	3	5	4	3	$h(3)$	4	0.16
5	4	5	1	4	$h(4)$	6	0.24
					$h(5)$	4	0.16
$M \times N = 25$					$h(6)$	1	0.04
					$h(7)$	6	0.24
					$n$	25	1



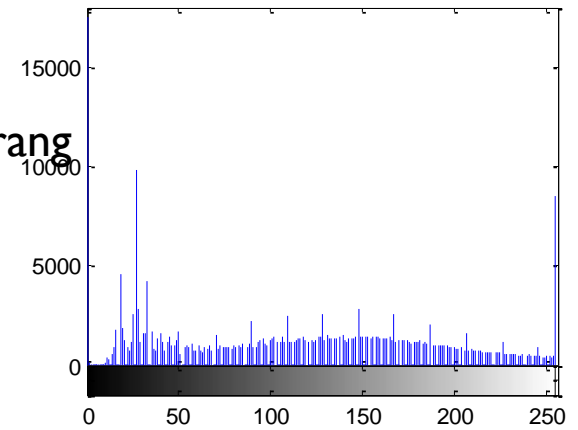




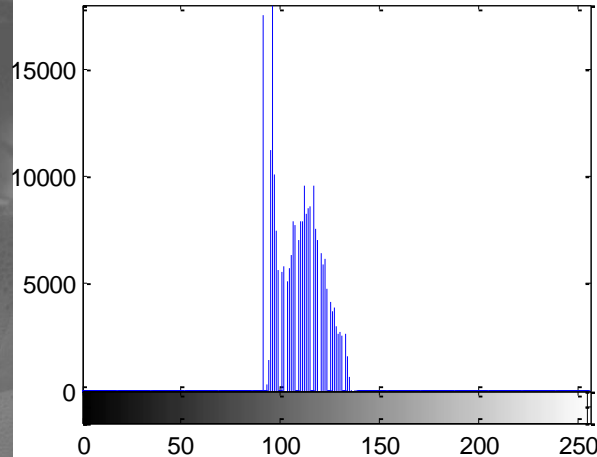
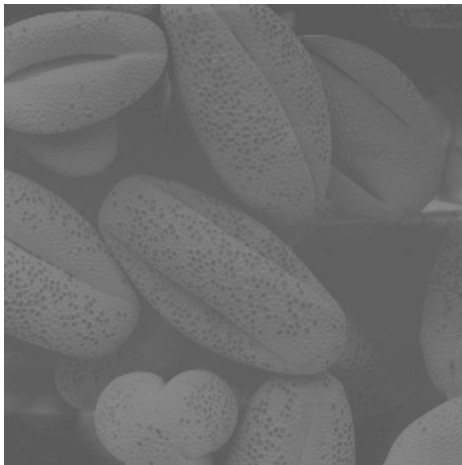
Citra gelap



Citra terang



Citra kontras tinggi



Citra kontras rendah

# HISTOGRAM NORMALIZATION

- Menskalakan nilai piksel secara linear untuk menggunakan secara penuh jangkauan yang gray-scale yang tersedia.
- Transformasi yang digunakan:

$$s_k = \frac{n_k - \min(n)}{\max(n) - \min(n)} x(L - 1)$$

untuk  $k=0,1,2,\dots,L-1$

disebut “histogram normalization”.

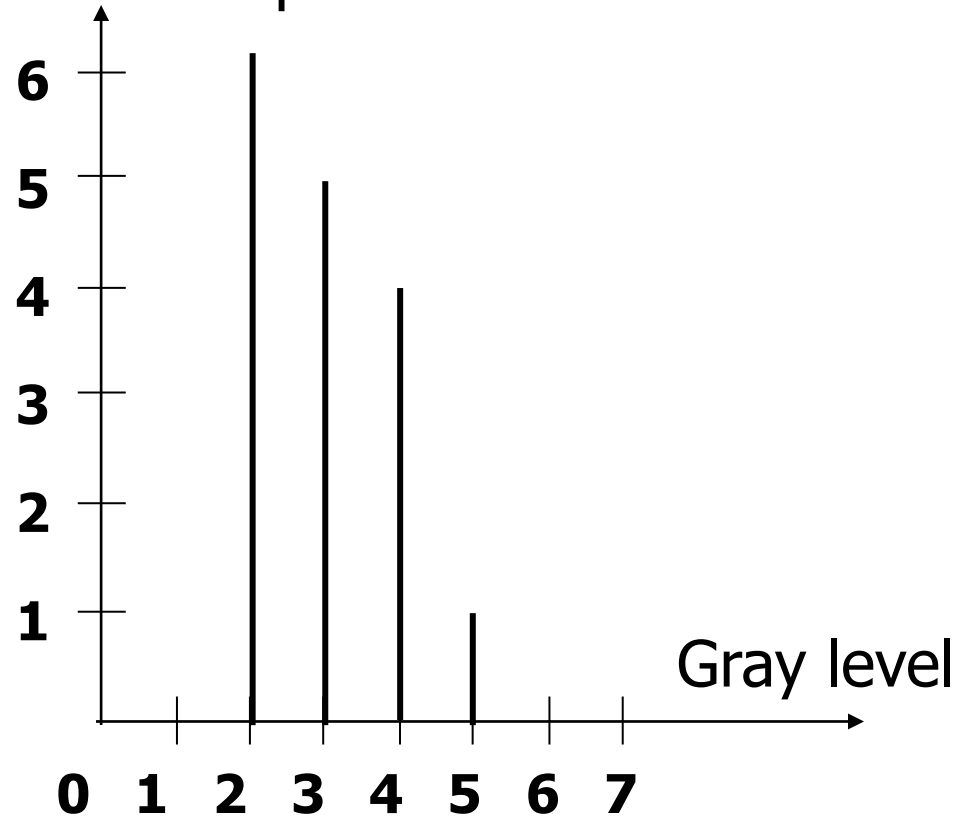
# CONTOH HISTOGRAM NORMALIZATION (I)

2	3	3	2
4	2	4	3
3	2	3	5
2	4	2	4

Input: 4x4 image

Gray scale =  $[0,7]$

Number of pixels



histogram

# CONTOH HISTOGRAM NORMALIZATION (2)

$$s_k = \frac{n_k - \min(n)}{\max(n) - \min(n)}$$

Gray Level(j)	0	1	2	3	4	5	6	7
$n_k$	0	0	6	5	4	1	0	0
	-	-	0	0.33 33	0.66 67	1	-	-
$s \times 7$	-	-	0	2.33 33 $\approx 2$	4.66 67 $\approx 5$	7	-	-

2	3	3	2
4	2	4	3
3	2	3	5
2	4	2	4

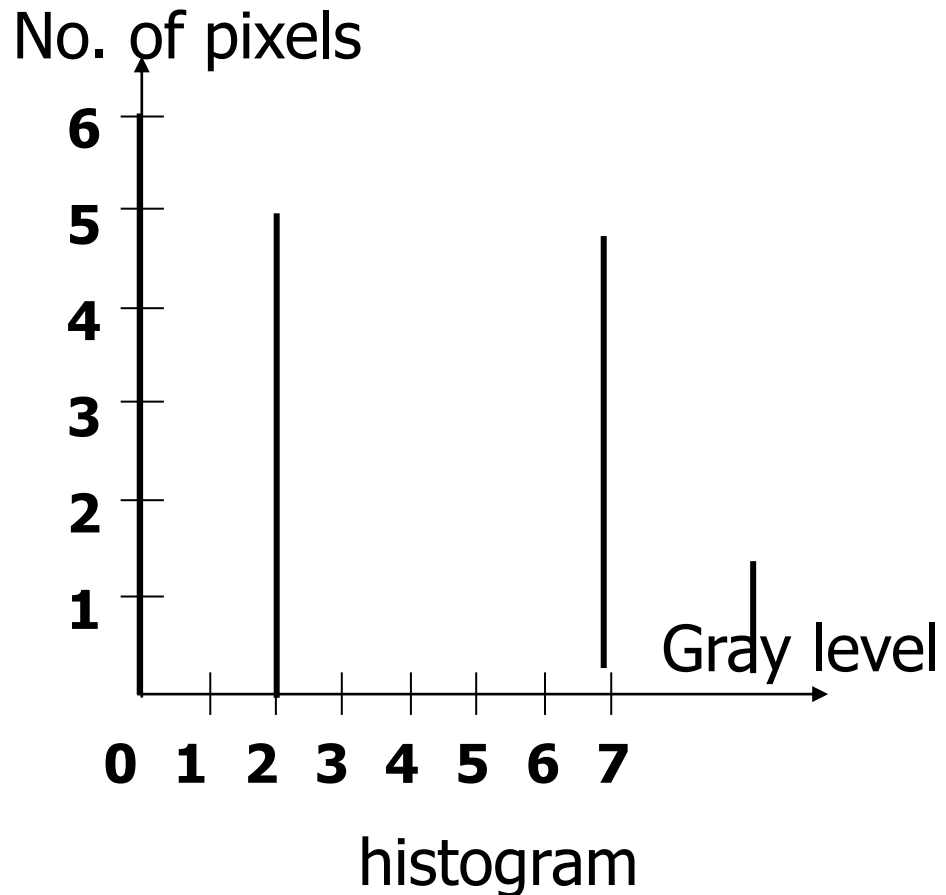


0	2	2	0
5	2	5	3
2	0	2	7
0	5	0	5

## CONTOH HISTOGRAM NORMALIZATION (3)

0	2	2	0
5	2	5	3
2	0	2	7
0	5	0	5

Output: 4x4 image  
Gray scale = [0,7]



# HISTOGRAM EQUALIZATION

- Histogram equalization digunakan untuk memetakan kembali nilai piksel untuk membuat perkiraan linear akumulasi histogram.
- Transformasi yang digunakan:

$$s_k = T(r_k) = \sum_{j=0}^k p_r(r_j)$$

$$s_k = T(r_k) = \sum_{j=0}^k \frac{n_j}{n}$$

untuk  $k=0,1,2,\dots,L-1$

disebut “histogram equalization” atau “histogram linearization”.

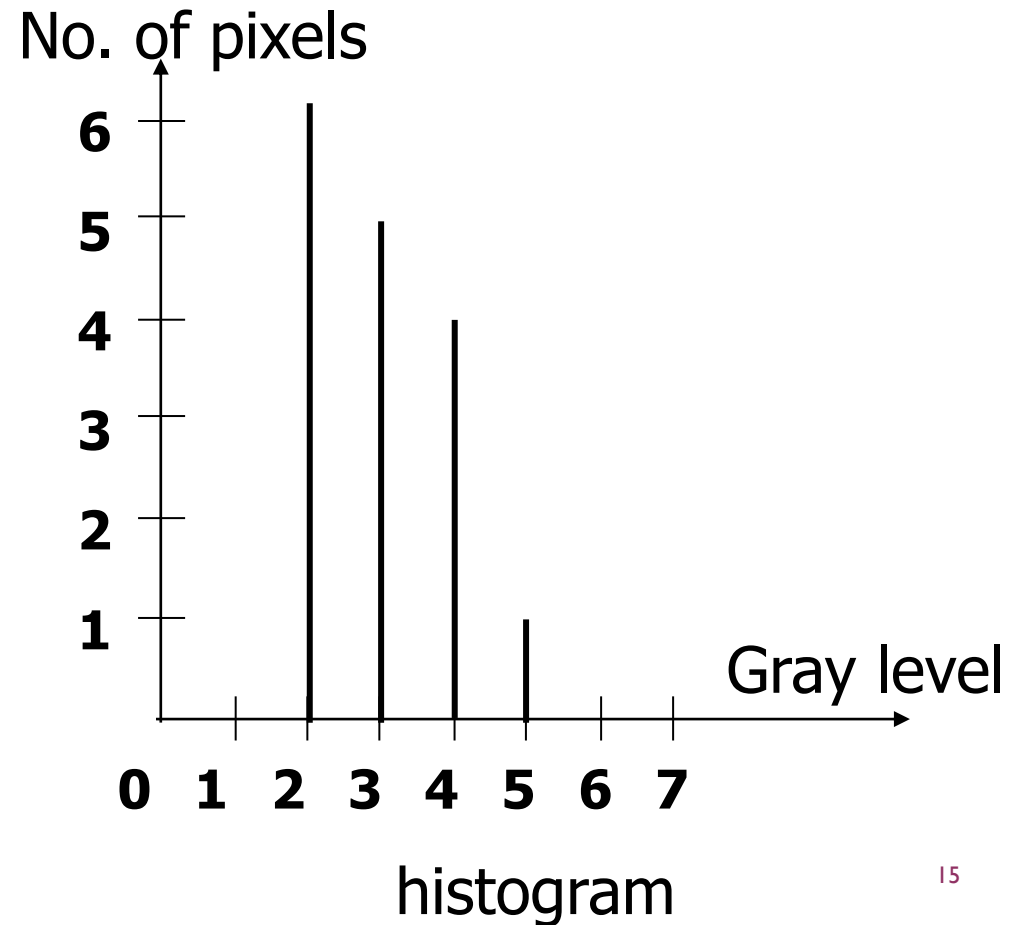
- Fungsi yang digunakan :
- $h = \text{histeq}(f, b)$ 
  - $h$  merupakan citra hasil equalisasi,
  - $f$  merupakan variabel citra dan  $b$  adalah jumlah bins yang digunakan dalam membentuk histogram. nilai default,  $b=64$  pada citra 6 bit.

# CONTOH HISTOGRAM EQUALIZATION (I)

2	3	3	2
4	2	4	3
3	2	3	5
2	4	2	4

Input: 4x4 image

Gray scale =  $[0,7]$



# CONTOH HISTOGRAM EQUALIZATION (2)

$$s = \sum_{j=0}^k \frac{n_j}{n}$$

Gray Level(j)	0	1	2	3	4	5	6	7
$n_j$	0	0	6	5	4	1	0	0
	0	0	6	11	15	16	16	16
	0	0	6 / 16	11/16	15/16	16/16	16/16	16/16
$s \times 7$	0	0	2.625 $\approx 3$	4.8125 $\approx 5$	6.5625 $\approx 7$	7	7	7

2	3	3	2
4	2	4	3
3	2	3	5
2	4	2	4



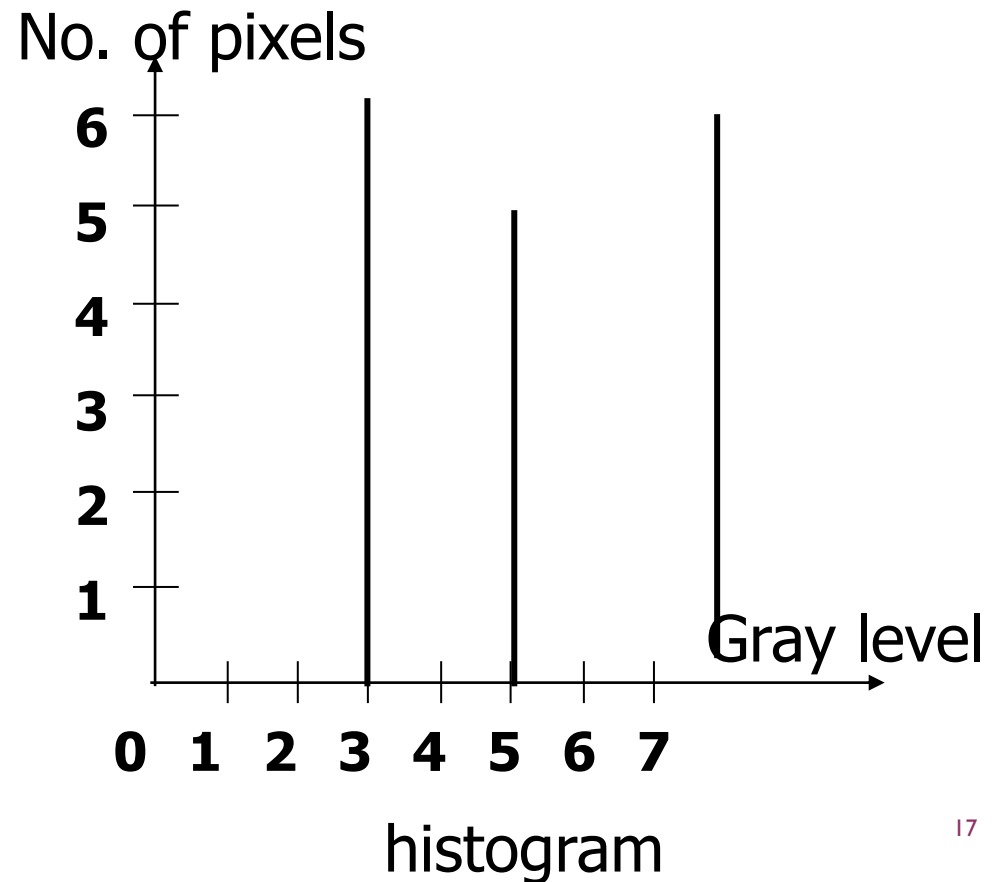
3	5	5	3
7	3	7	5
5	3	5	7
3	7	3	7



## CONTOH HISTOGRAM EQUALIZATION (3)

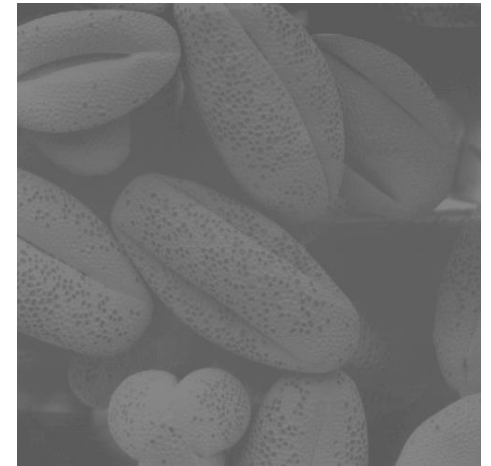
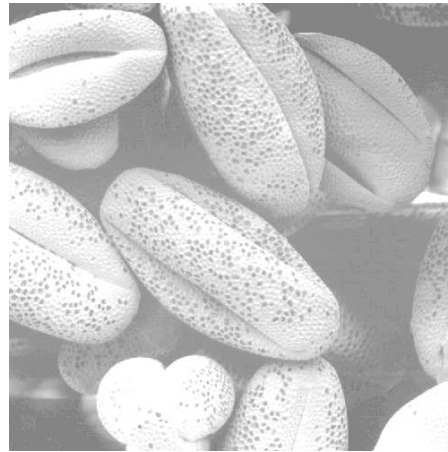
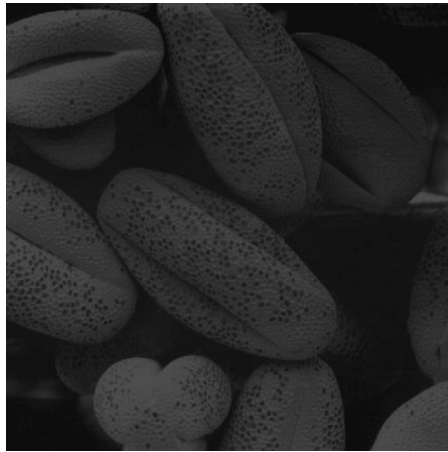
3	5	5	3
7	3	7	5
5	3	5	7
3	7	3	7

Output: 4x4 image  
Gray scale = [0,7]

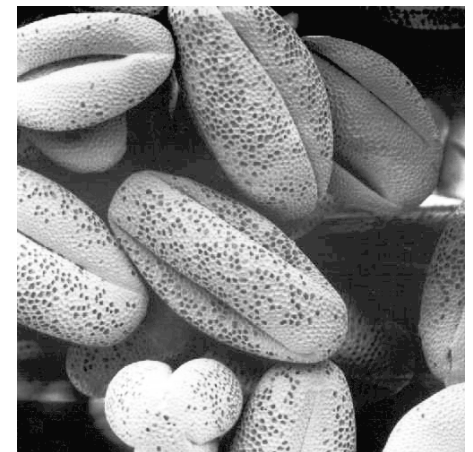
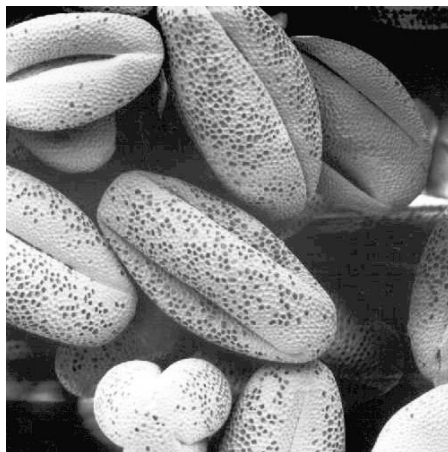


# CONTOH HISTOGRAM EQUALIZATION (4)

sebelum

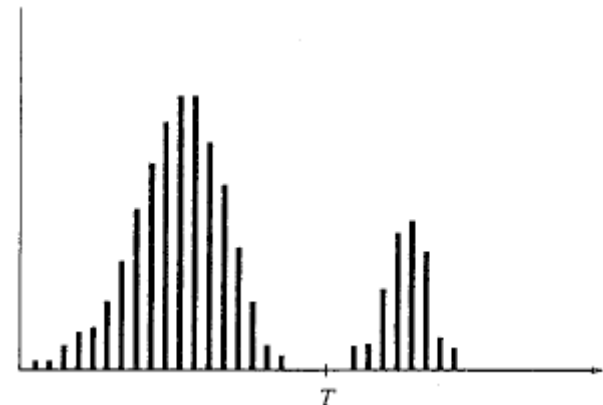
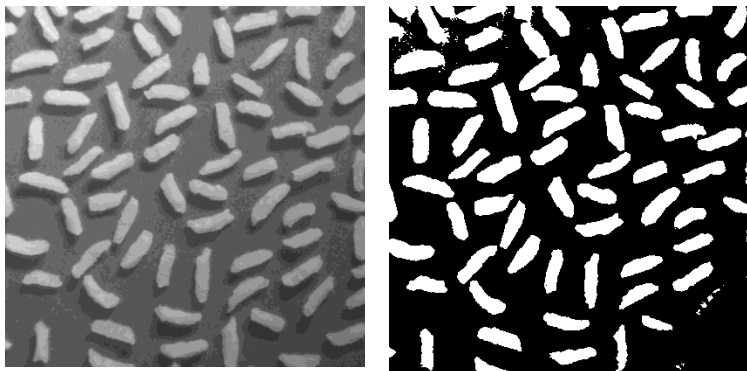


sesudah



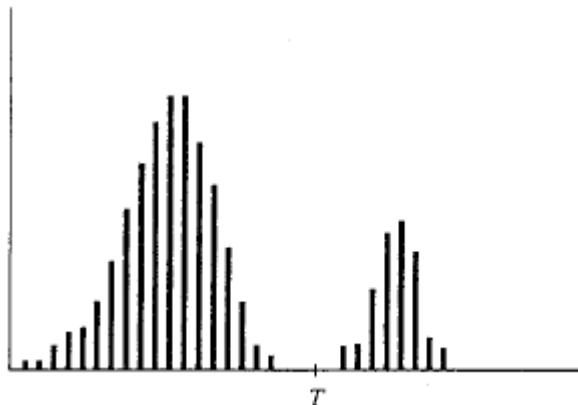
# CITRA BINER

- Citra yang hanya mempunyai nilai level intensitas keabuan hitam dan putih (0 dan 1).
- Menjadi dasar dalam pengolahan citra: morfologi, segmentasi, representasi, pengenalan pola, dsb.
- Formula:
$$g(x, y) = \begin{cases} 1 & \text{jika } f(x, y) \geq T \\ 0 & \text{jika } f(x, y) < T \end{cases}$$
- Piksel yang diberi nilai 1 berkaitan dengan obyek sedangkan piksel yang diberi nilai 0 berkaitan dengan background.
- Ketika T adalah konstanta, pendekatan ini disebut *global thresholding*.



# THRESHOLDING

- Salah satu cara memilih thresholding adalah dengan pemeriksaan visual histogram citra.
- Histogram dalam gambar secara jelas mempunyai dua mode yang berbeda.
  - Mudah untuk memilih threshold  $T$  yang membaginya.
- Metode yang lain dalam pemilihan  $T$  adalah dengan *trial and error*, mengambil beberapa threshold berbeda sampai satu nilai  $T$  yang memberikan hasil yang baik sebagai keputusan yang ditemukan observer .



## THRESHOLDING (2)

- Prosedur untuk pemilihan threshold secara otomatis:
  1. Memilih perkiraan awal  $T$ . Disarankan perkiraan awal adalah titik tengah antara nilai intensitas minimum dan maksimum dalam citra.
  2. Mensegmentasi citra menggunakan  $T$ . Ini akan menghasilkan dua kelompok piksel:  $G_1$ , yang berisi semua piksel dengan nilai intensitas  $\geq T$ , dan  $G_2$ , yang berisi semua piksel dengan nilai intensitas  $< T$ .
  3. Menghitung rata-rata nilai intensitas  $\mu_1$  dan  $\mu_2$  untuk piksel-piksel dalam region  $G_1$  dan  $G_2$ .
  4. Menghitung nilai threshold yang baru:

$$T = \frac{1}{2}(\mu_1 + \mu_2)$$

5. Mengulangi langkah 2 sampai 4 sampai perbedaan  $T$  dalam iterasi yang berturut-turut lebih kecil daripada parameter  $T_0$  sebelumnya.

# FORMULASI OTSU

- Normalisasi histogram sebagai fungsi probability discrete density, sebagai:

- $$p_r(r_q) = \frac{n_q}{n} \quad q = 0, 1, 2, \dots, L - 1$$

- di mana  $n$  adalah total jumlah piksel dalam citra,  $n_q$  adalah jumlah piksel yang dipunyai level intensitas  $r_q$ , dan  $L$  adalah total jumlah level intensitas yang tersedia dalam citra.
- Andaikan bahwa threshold  $k$  dipilih maka  $C_0$  adalah sekumpulan piksel dengan level  $[0, 1, \dots, k - 1]$  dan  $C_1$  adalah sekumpulan piksel dengan level  $[k, k + 1, \dots, L - 1]$ .
- Metode Otsu memilih nilai  $k$  yang memaksimalkan *between-class variance*  $\sigma_B^2$ , yang didefinisikan sebagai:

$$\sigma_B^2 = \omega_0 (\mu_0 - \mu_T)^2 + \omega_1 (\mu_1 - \mu_T)^2$$

$$\omega_0 = \sum_{q=0}^{k-1} p_q(r_q) \quad \mu_0 = \sum_{q=0}^{k-1} qp_q(r_q) / \omega_0$$

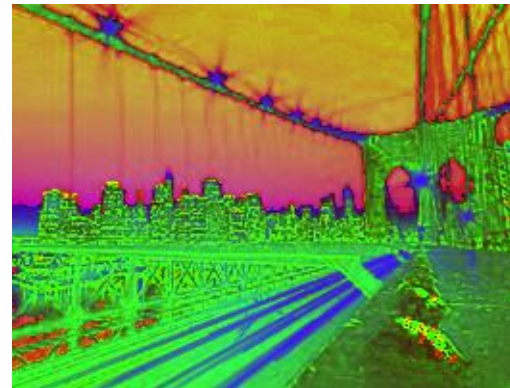
$$\omega_1 = \sum_{q=k}^{L-1} p_q(r_q) \quad \mu_1 = \sum_{q=k}^{L-1} qp_q(r_q) / \omega_1$$

$$\mu_T = \sum_{q=0}^{L-1} qp_q(r_q)$$

# MODEL WARNA HSI



Citra RGB



Citra HSI



Citra Hue



Citra Saturation



Citra Intensity/Gray

# KONVERSI WARNA RGB KE HSI

Hue

$$H = \begin{cases} \theta & \text{Jika } B \leq G \\ 360 & \text{Jika } B > G \end{cases}$$

$$\theta = \cos^{-1} \left\{ \frac{\frac{1}{2}[(R-G) + (R-B)]}{[(R-G)^2 + (R-B)(G-B)]^{1/2}} \right\}$$

Saturation

$$S = 1 - \frac{3}{(R+G+B)} [\min(R, G, B)]$$

Intensity / Gray

$$I = \frac{1}{3}(R+G+B)$$



# KONVERSI WARNA HSI KE RGB

- Jika diberikan nilai HSI dalam interval  $[0, 1]$  maka dapat dicari nilai RGB dalam range yang sama.
- Persamaan yang aplikatif tergantung pada nilai H.
- Ada tiga sektor yang menarik, yang berhubungan dengan interval  $120^\circ$  dalam pembagian warna primer.
  - Dimulai dengan perkalian H dengan  $360^\circ$ , yang mengembalikan hue ke nilai aslinya dalam range  $[0^\circ, 360^\circ]$

Sektor RG ( $0^\circ \leq H < 120^\circ$ )

$$B = I(1 - S)$$

$$R = I \left[ 1 + \frac{S \cos H}{\cos(60^\circ - H)} \right]$$

$$G = 3I - (R + B)$$

Sektor GB ( $120^\circ \leq H < 240^\circ$ )

$$H = H - 120^\circ$$

Kemudian komponen RGB dihitung:

$$R = I(1 - S)$$

$$G = I \left[ 1 + \frac{S \cos H}{\cos(60^\circ - H)} \right]$$

$$B = 3I - (R + G)$$

Sektor BR ( $240^\circ \leq H \leq 360^\circ$ )

$$H = H - 240^\circ$$

Komponen RGB dihitung dengan:

$$G = I(1 - S)$$

$$B = I \left[ 1 + \frac{S \cos H}{\cos(60^\circ - H)} \right]$$

$$R = 3I - (G + B)$$