

09 - Image Enhancement

(Bagian 2)

IF4073 Interpretasi dan Pengolahan Citra

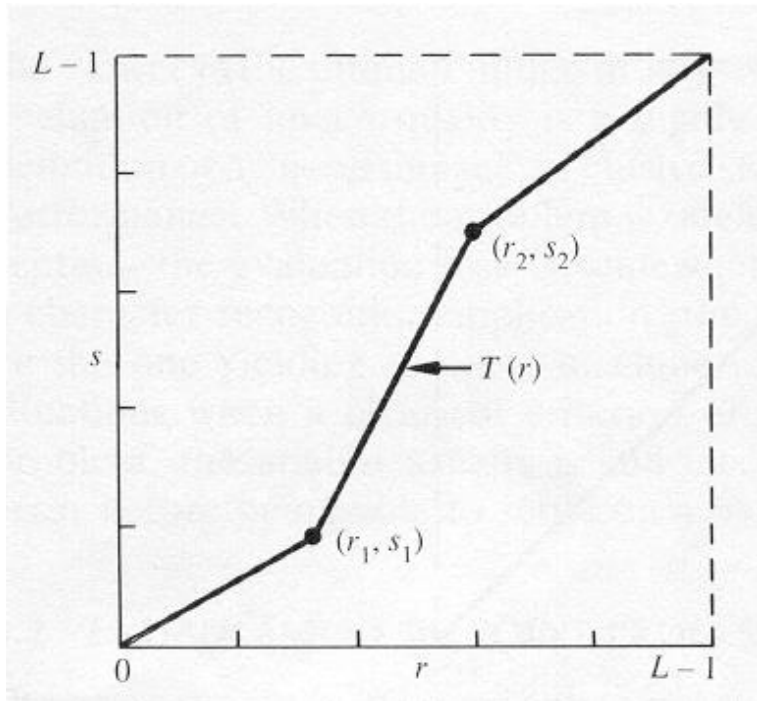
Oleh: Rinaldi Munir



Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2021

Peregangan Kontras (revisited)

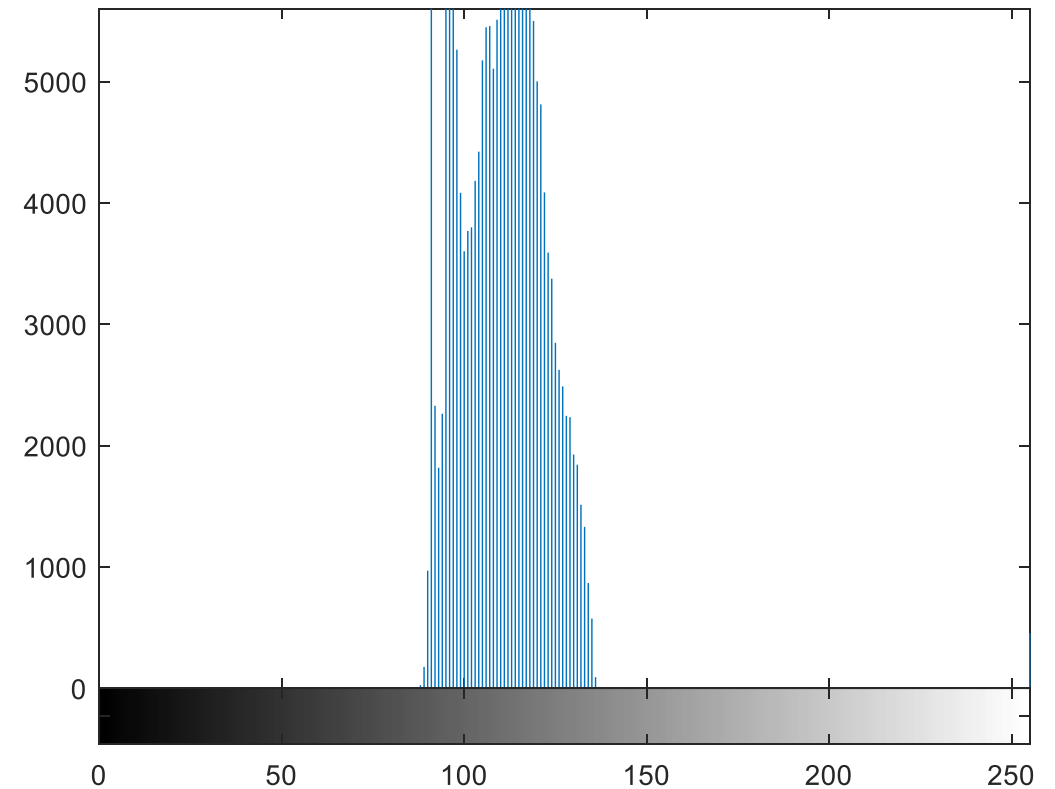
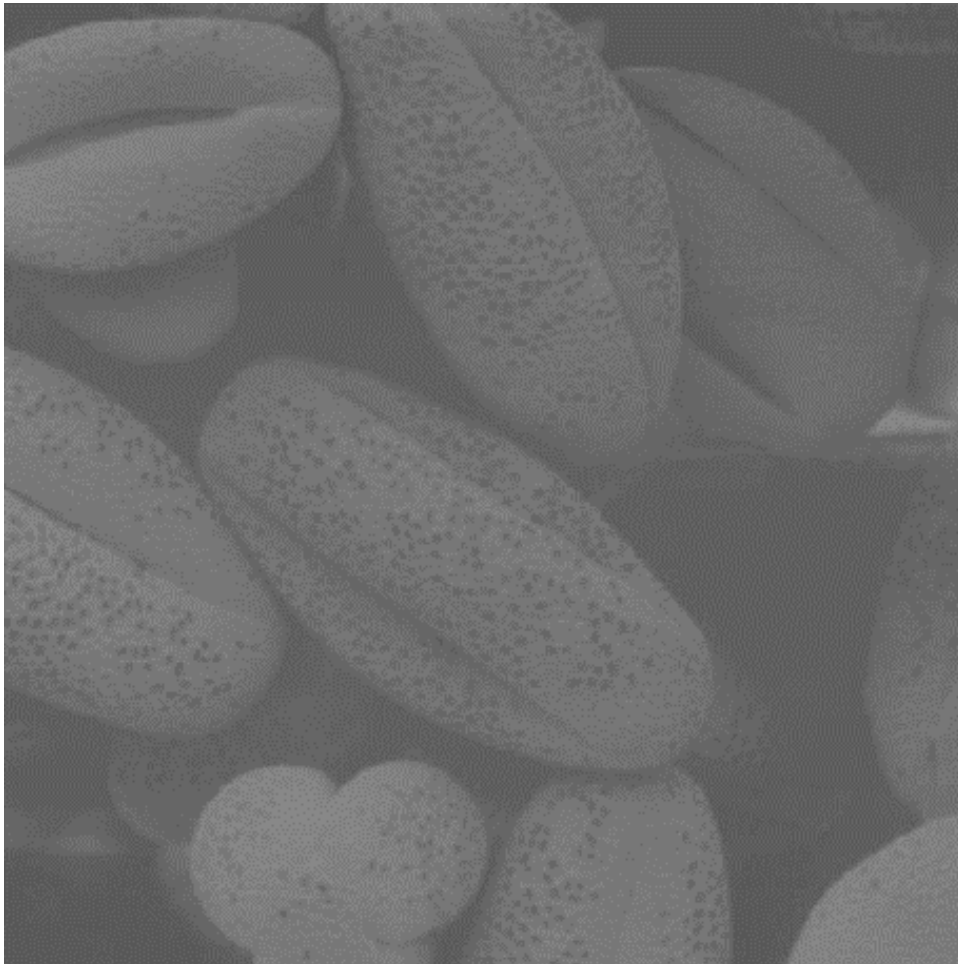
- Tinjau kembali topik peregangan kontras (*contrast stretching*).
- Peregangan kontras termasuk ke dalam fungsi transformasi sepotong-sepotong (*piece-wise linear transformation function*)



Fungsi transformasi linier sepotong-sepotong:

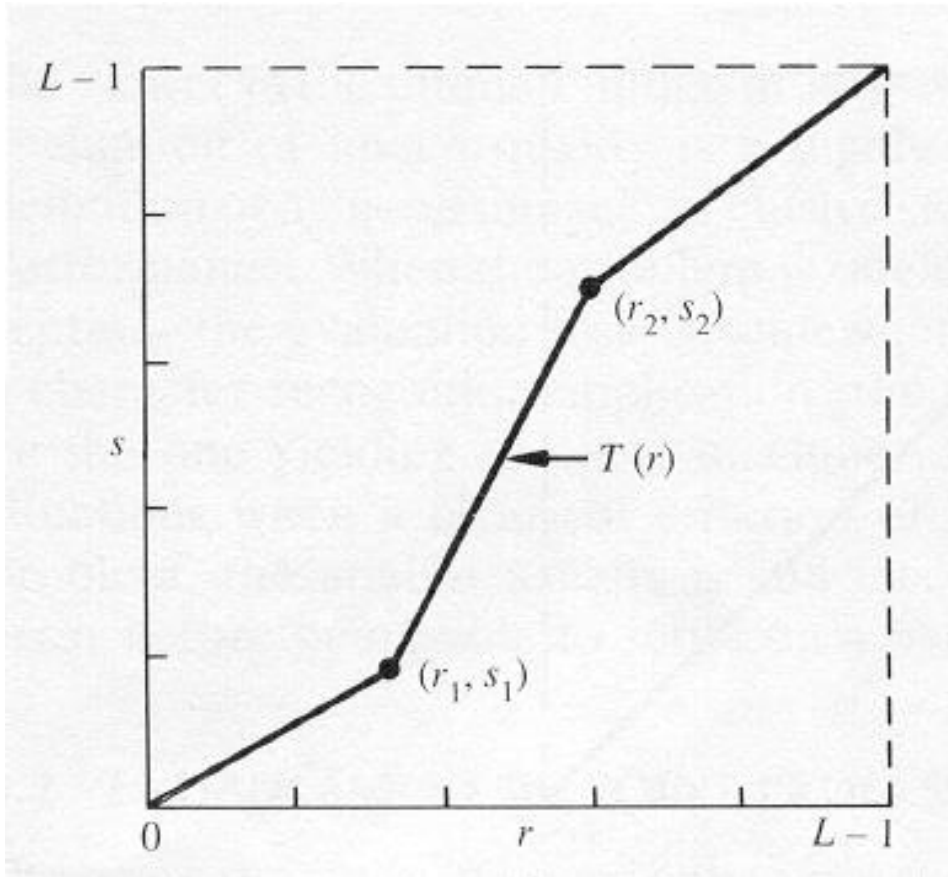
1. *Contrast stretching*
2. *Gray-level slicing*
3. *Bit-plane slicing*

- Peregangan kontras merupakan metode sederhana untuk memperbaiki citra yang memiliki kontras rendah



Ciri-ciri citra kontras-rendah: histogram sempit menumpuk di tengah

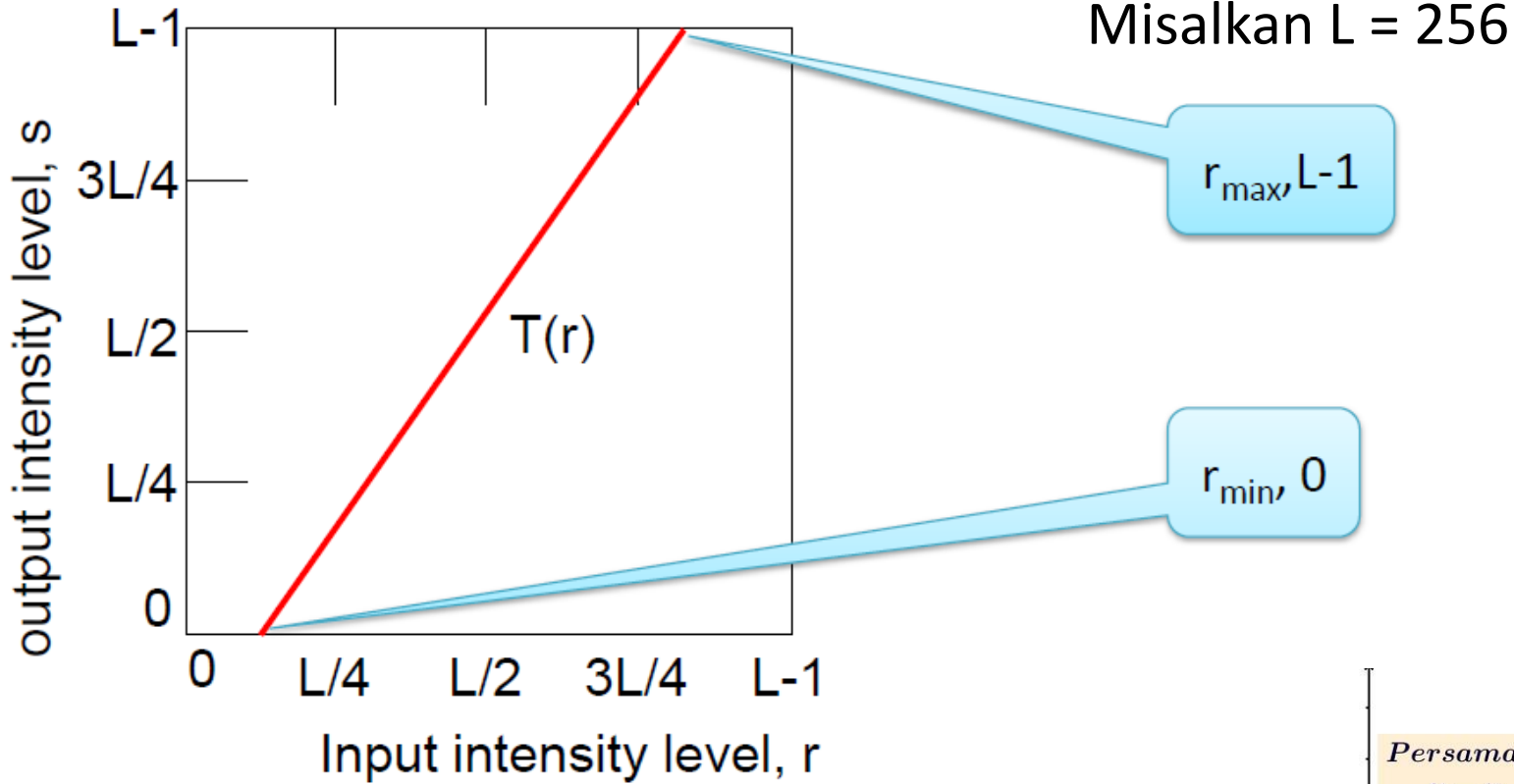
- Bagaimana menentukan r_1 dan r_2 ?



Salah satu pendekatan:

- cari pixel bernilai minimum, misalkan $rmin$
- cari pixel bernilai maksimum, misalkan $rmax$
- pixel-pixel di bawah $rmin$ diset 0
- Pixel-pixel di atas $rmax$ diset $L - 1$
- $r1 = rmin, r2 = rmax$
- tentukan persamaan garis yang menghubungkan titik $(rmin, 0)$ dan $(rmax, L - 1)$ dengan persamaan umum garis: $(y - y1)/(y2 - y1) = (x - x1)/(x2 - x1)$
- Petakan nilai keabuan yang lain di antara $(rmin, 0)$ dan $(rmax, L - 1)$ dengan menggunakan persamaan tersebut

$$(r_1, s_1) = (r_{\min}, 0) \text{ and } (r_2, s_2) = (r_{\max}, L-1)$$

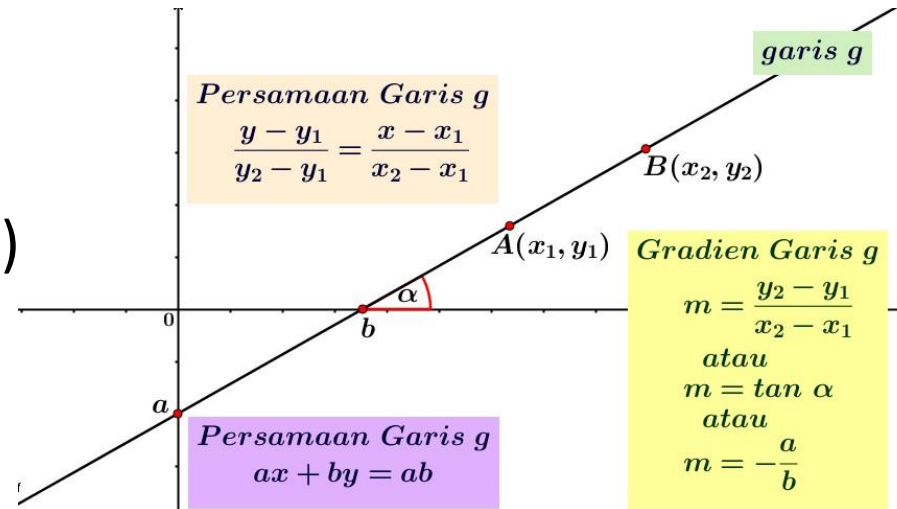


Persamaan garis yang melalui $(r_{\min}, 0)$ dan $(r_{\max}, 255)$

$$(s - 0) / (255 - 0) = (r - r_{\min}) / (r_{\max} - r_{\min})$$

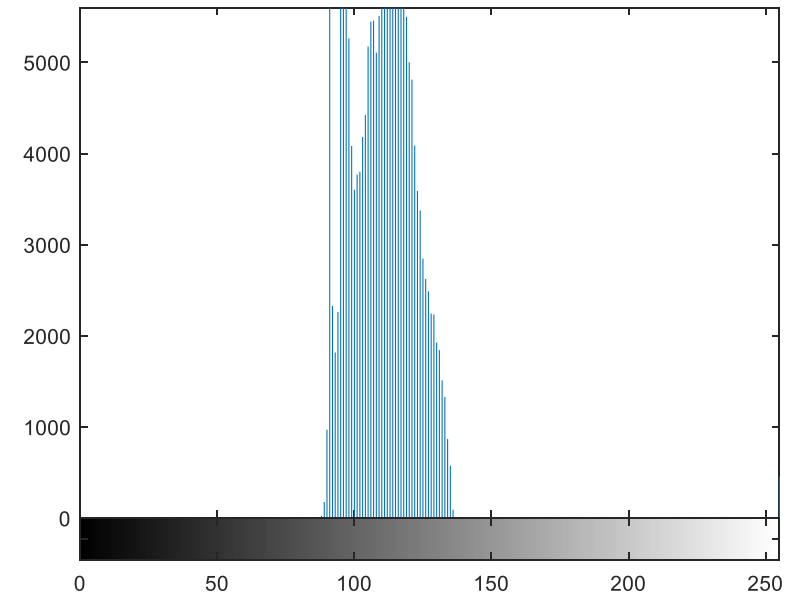
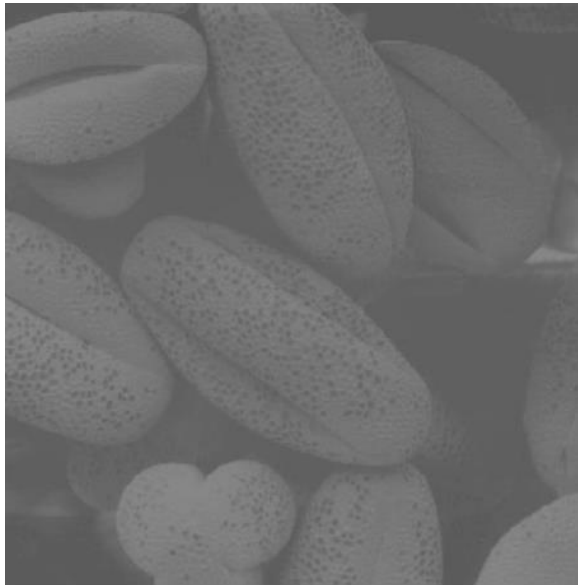
$$s = 255(r - r_{\min}) / (r_{\max} - r_{\min})$$

s



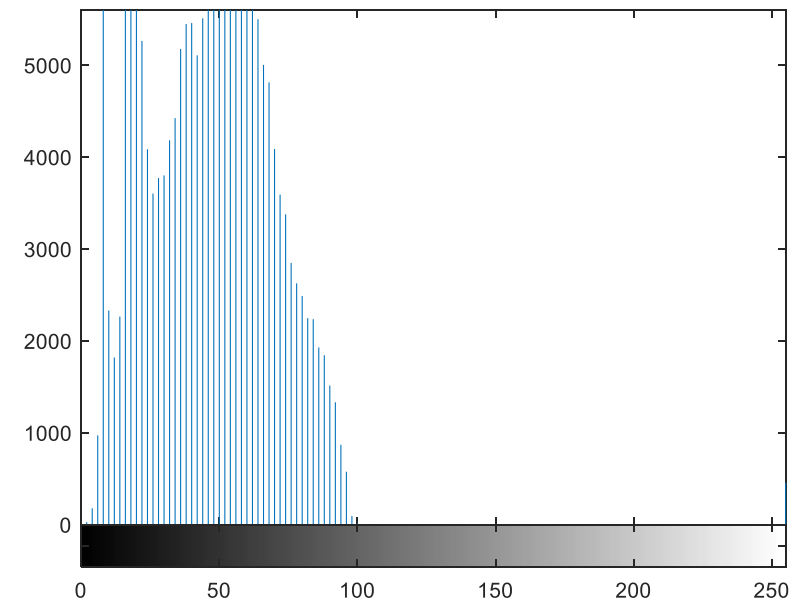
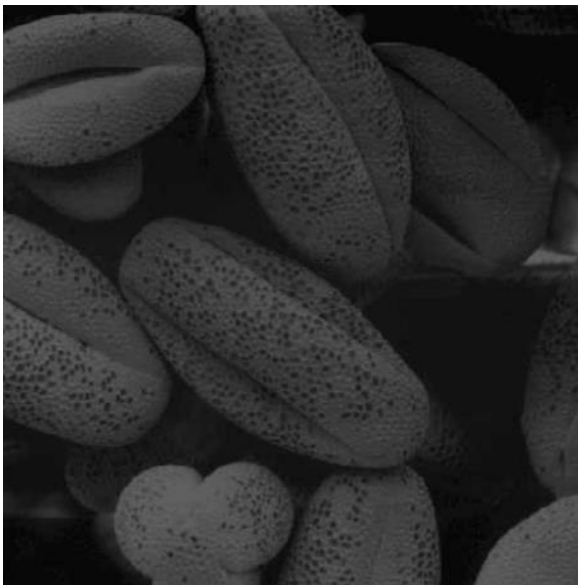
```
clear all;
clc;
I = imread('image1.bmp'); %read the image
rmin = min(I(:)); % find the min. value of pixel in the image
rmax = max(I(:)); % find the max. value of pixel in the image
I_new = (I - rmin).*(255/(rmax - rmin)); % transform the image
figure,imshow(I); % display original image
figure,imhist(I); % display histogram of original image
figure,imshow(I_new); % display transformed image
figure,imhist(I_new); % display histogram of transformed image
```

Sebelum



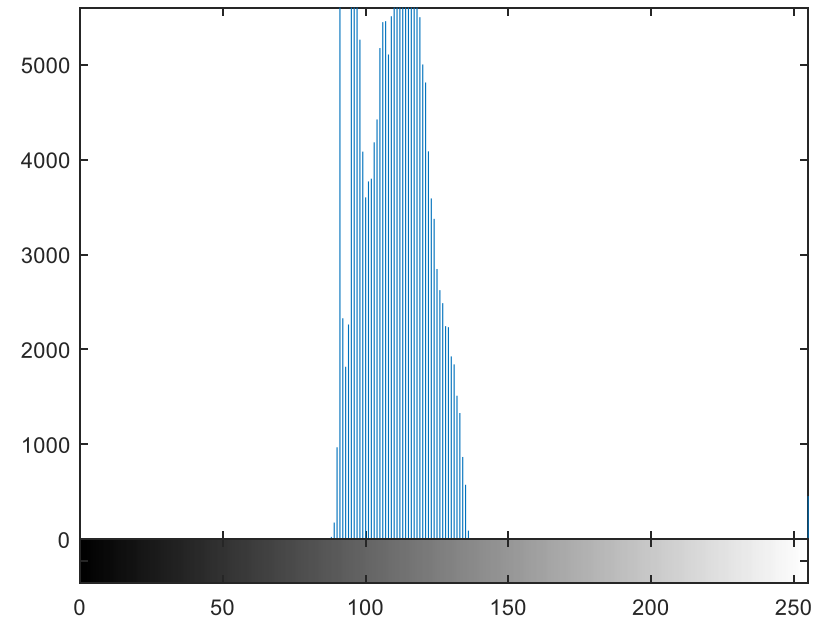
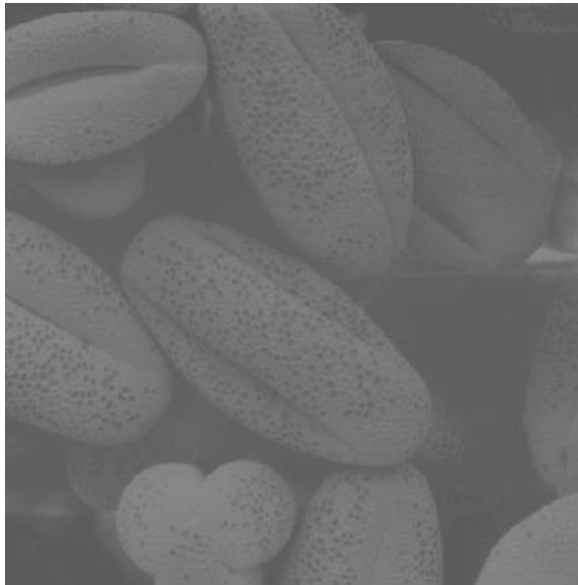
$$r_{\min} = 87$$
$$r_{\max} = 255$$

Sesudah

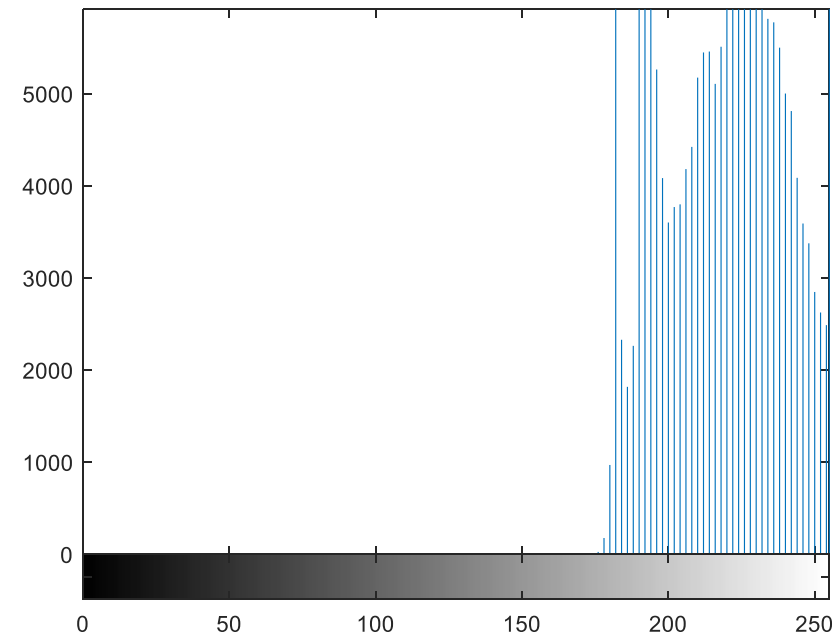
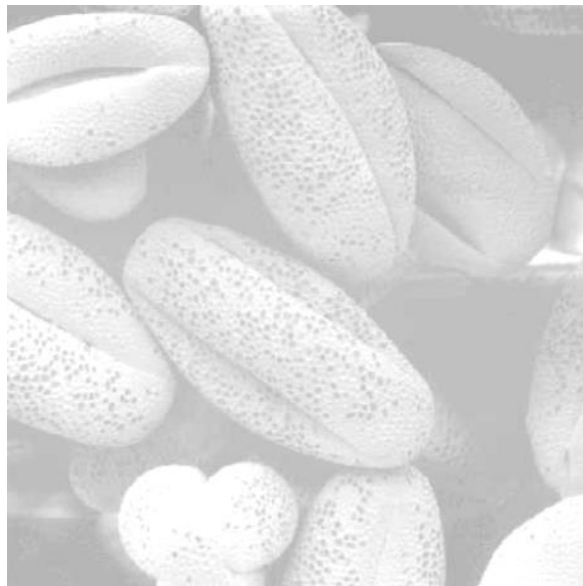


```
clear all;
clc;
I = imread('image1.bmp'); %read the image
rmin = min(I(:)); % find the min. value of pixel in the image
rmax = max(I(:)); % find the max. value of pixel in the image
m = 255/(rmax - rmin); % find the slope of line joining point (0,255) to
(rmin,rmax)
c = 255 - m*rmax; % find the intercept of the straight line with the
axis
I_new = m*I + c; % transform the image according to new slope
figure,imhist(I); % display histogram of original image
figure,imshow(I_new); % display transformed image
figure,imhist(I_new); % display histogram of transformed image
```


Sebelum

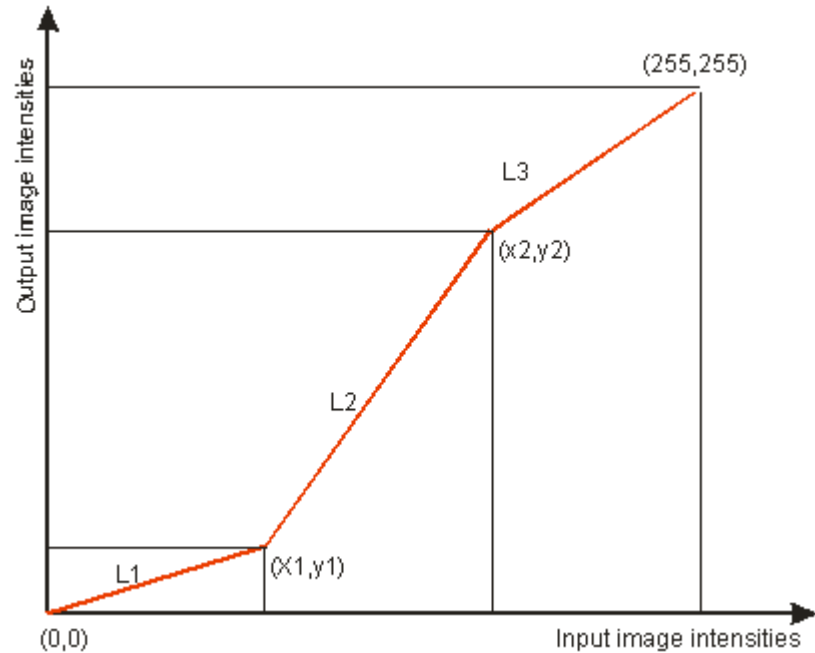


Sesudah



$r_{\min} = 87$
 $r_{\max} = 255$

- Alternatif lain fungsi transformasi sepotong-sepotong:



$$y = \begin{cases} \frac{y_1}{x_1}x, & 0 \leq x \leq x_1 \\ \frac{y_2 - y_1}{x_2 - x_1}x + y_1, & x_1 < x < x_2 \\ \frac{255 - y_2}{255 - x_2}x + y_2, & x_2 < x < 255 \end{cases}$$

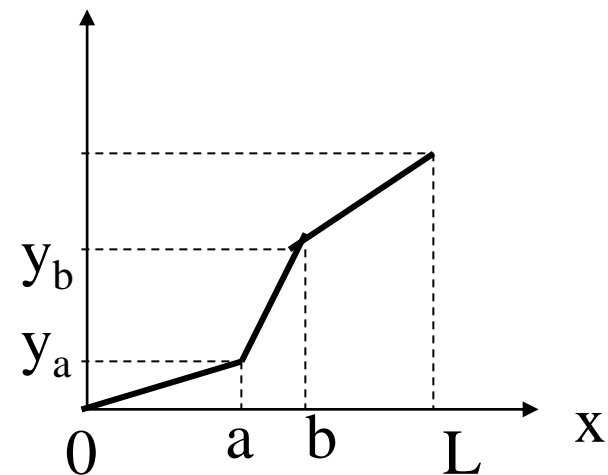


$$y = \begin{cases} \alpha x & 0 \leq x < a \\ \beta(x - a) + y_a & a \leq x < b \\ \gamma(x - b) + y_b & b \leq x < L \end{cases}$$

Persamaan garis L1: $y = \frac{y_1}{x_1}x$

Persamaan garis L2: $y = \frac{y_2 - y_1}{x_2 - x_1} \cdot x + y_1$

Persamaan garis L3: $y = \frac{255 - y_2}{255 - x_2} \cdot x + y_2$

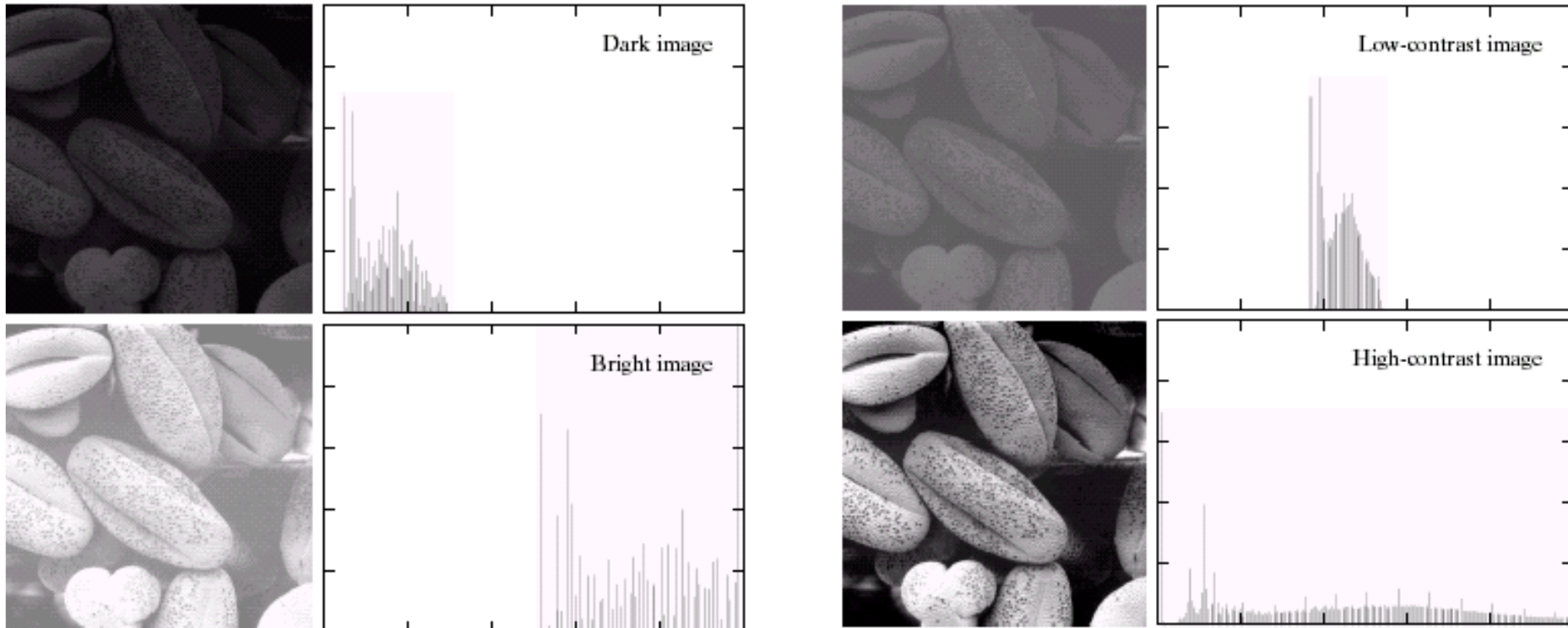




$$a = 50, b = 150, \alpha = 0.2, \beta = 2, \gamma = 1, y_a = 30, y_b = 200$$

Histogram Enhancement

- Histogram memberikan informasi tentang kualitas citra.
- Citra gelap, citra terang, dan yang memiliki kontras rendah memiliki histogram yang tidak merata penyebarannya.



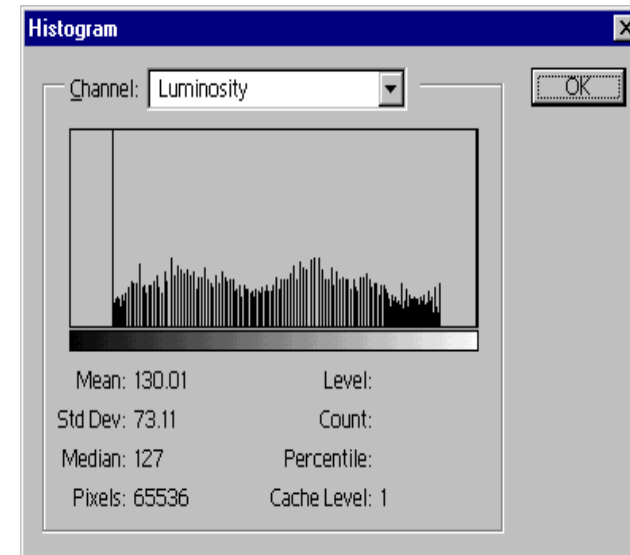
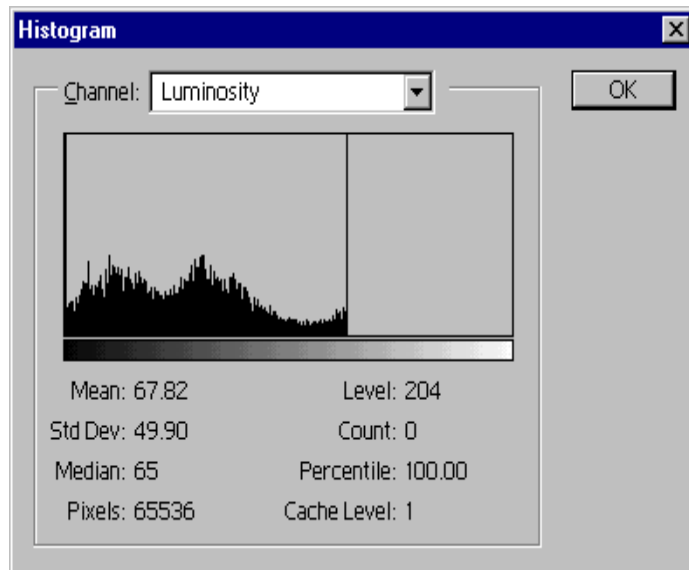
- Agar dapat memiliki histogram citra sesuai dengan keinginan kita, maka penyebaran nilai-nilai intensitas pada citra harus diubah.
- Dua metode pengubahan citra berdasarkan histogram:
 1. **Perataan histogram (*histogram equalization*)**

Nilai-nilai intensitas di dalam citra diubah sehingga penyebarannya seragam (*uniform*).
 2. **Spesifikasi histogram (*histogram specification*)**

Nilai-nilai intensitas di dalam citra diubah agar diperoleh histogram dengan bentuk yang dispesifikasikan oleh pengguna.

1. Perataan Histogram

- *Histogram equalization*
- Tujuan: untuk memperoleh penyebaran histogram yang merata, sedemikian sehingga setiap derajat keabuan memiliki jumlah *pixel* yang relatif sama.
- Memperlebar rentang nilai keabuan, sehingga dapat meningkatkan kekontrasan citra.
- Termasuk operasi dalam aras global

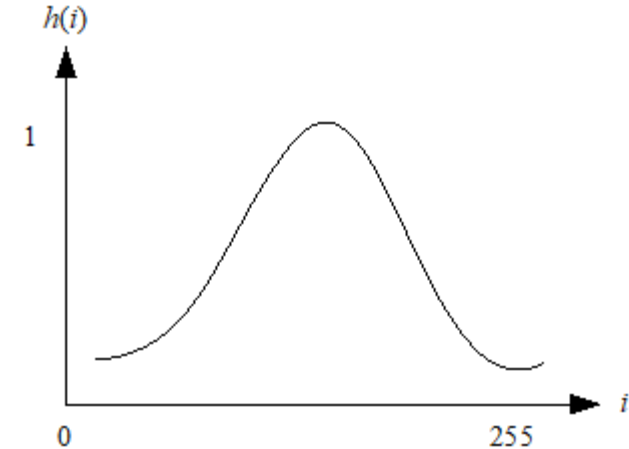


- Ingat kembali histogram ternormalisasi $h_k = \frac{n_k}{n}$ menyatakan peluang *pixel* dengan derajat keabuan tertentu:

$$P_r(r_k) = \frac{n_k}{n}$$

n_k = jumlah *pixel* dengan nilai keabuan r_k

n = jumlah seluruh *pixel* di dalam citra



- Misalkan nilai keabuan r juga dinormalkan terhadap nilai keabuan maksimum:

$$r_k = \frac{k}{L-1} \quad , 0 \leq k \leq L-1$$

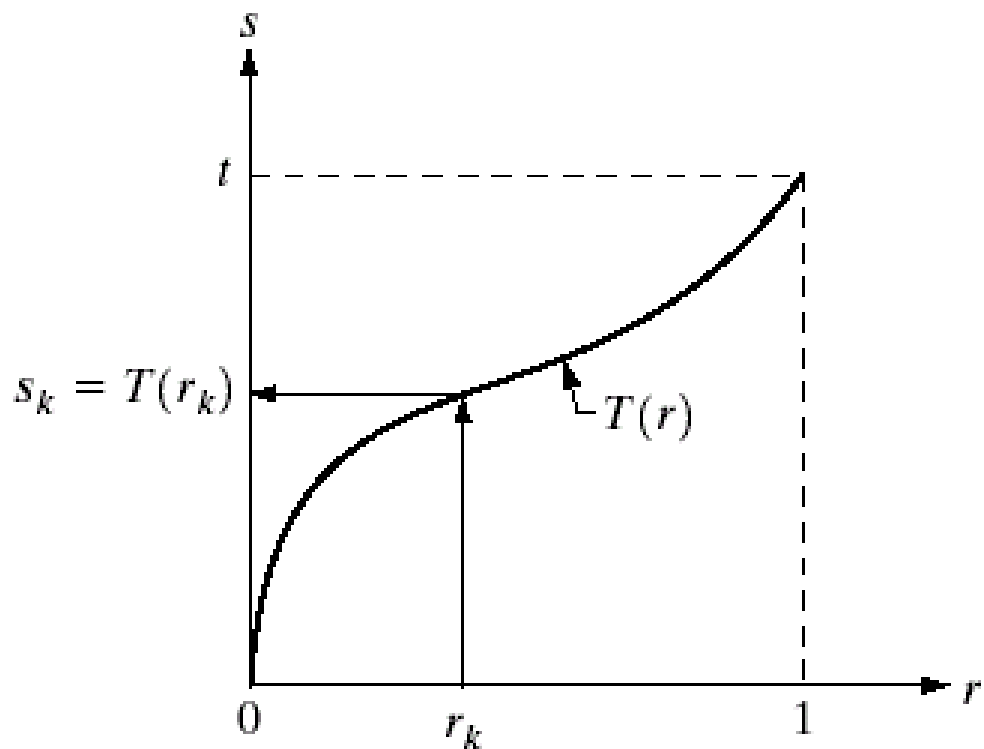
Contoh: Jika $L = 8$, maka nilai-nilai r_k dinyatakan di dalam tabel 1.

Tabel 1 Nilai-nilai r_k jika $L = 8$

k	r_k
0	$0/7 = 0$
1	$1/7$
2	$2/7$
3	$3/7$
4	$4/7$
5	$5/7$
6	$6/7$
7	$7/7 = 1$

- Perataan histogram artinya mengubah derajat keabuan r dengan derajat keabuan yang baru, s , dengan fungsi transformasi T :

$$s = T(r).$$



Sifat:

- (1) $T(r)$ adalah fungsi bernilai tunggal (*single-value*) yang naik monoton di dalam selang $0 \leq r \leq 1$;
- (2) $0 \leq T(r) \leq 1$ untuk $0 \leq r \leq 1$
- (3) $r = T^{-1}(s)$ untuk $0 \leq s \leq 1$

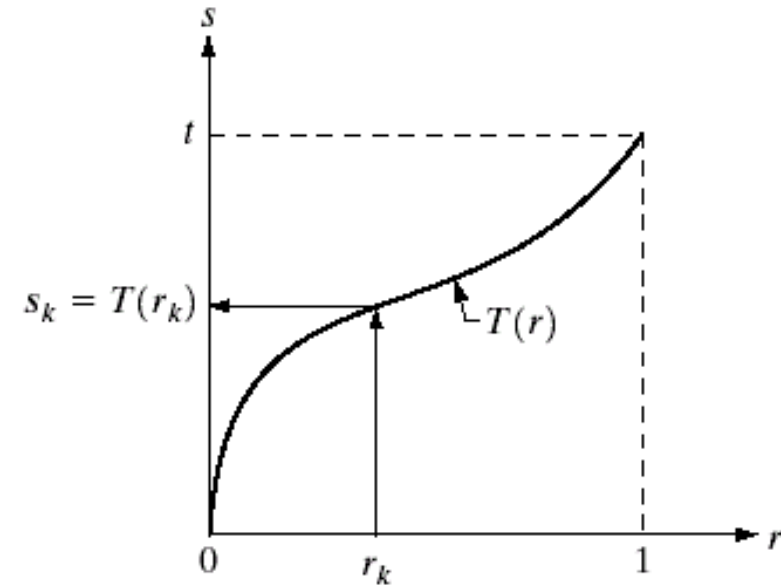
- Untuk fungsi histogram yang menerus:

$$s = T(r) = \int_0^r P_r(w)dw \quad , 0 \leq r \leq 1$$

- Dalam bentuk diskrit:

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

$$0 \leq r_k \leq 1, \quad k = 0, 1, 2, \dots, L-1$$

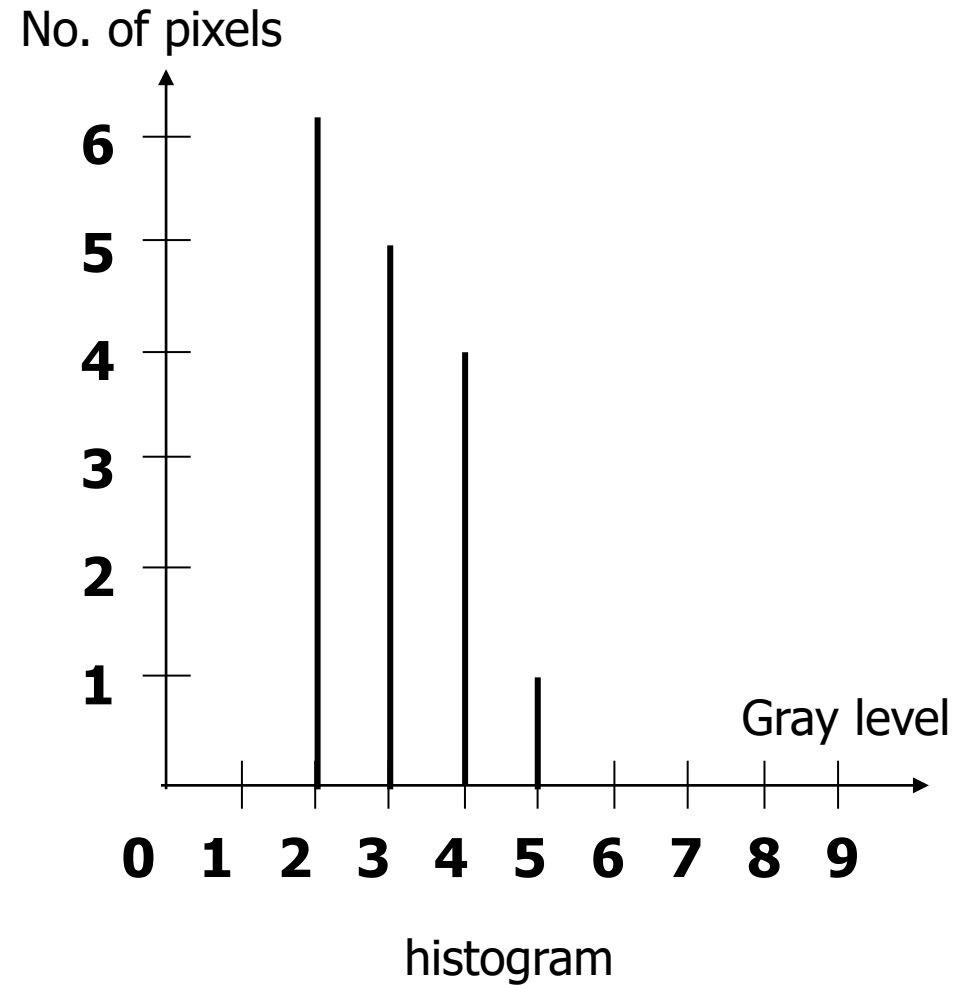


Contoh:

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

4x4 image

Gray scale = [0,9]



Sumber: Ali Javed, Digital Image Processing, Chapter # 3, Image Enhancement in Spatial Domain

Gray Level (j)	0	1	2	3	4	5	6	7	8	9
No. of pixels	0	0	6	5	4	1	0	0	0	0
$\frac{n_j}{n}$	0	0	6/16	5/16	4/16	1/16	0	0	0	0
$s = \sum_{j=0}^k \frac{n_j}{n}$	0	0	6 / 16	11/16	15/16	16/16	16/16	16/16	16/16	16/16
$s \times 9$	0	0	3.3 ≈ 3	6.1 ≈ 6	8.4 ≈ 8	9	9	9	9	9

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

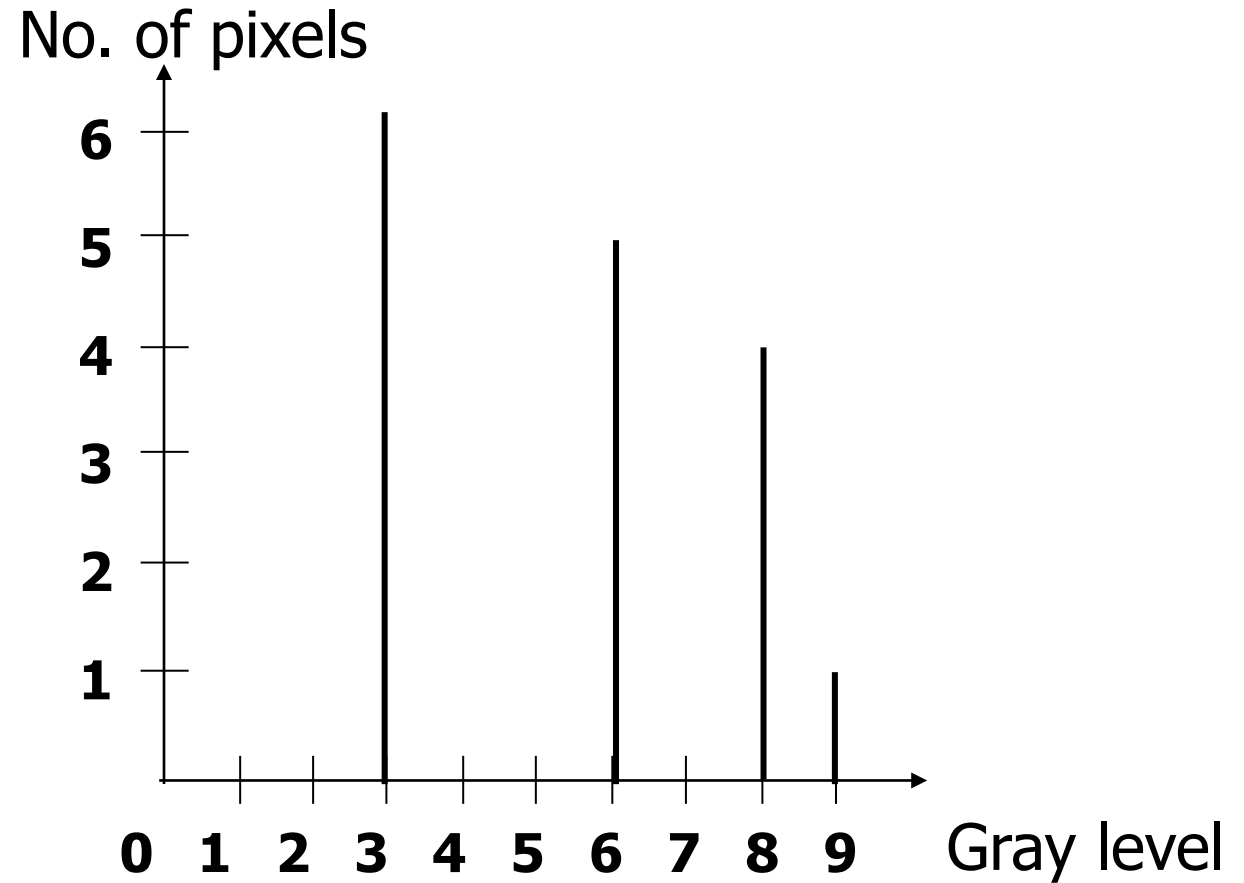


3	6	6	3
8	3	8	6
6	3	6	9
3	8	3	8

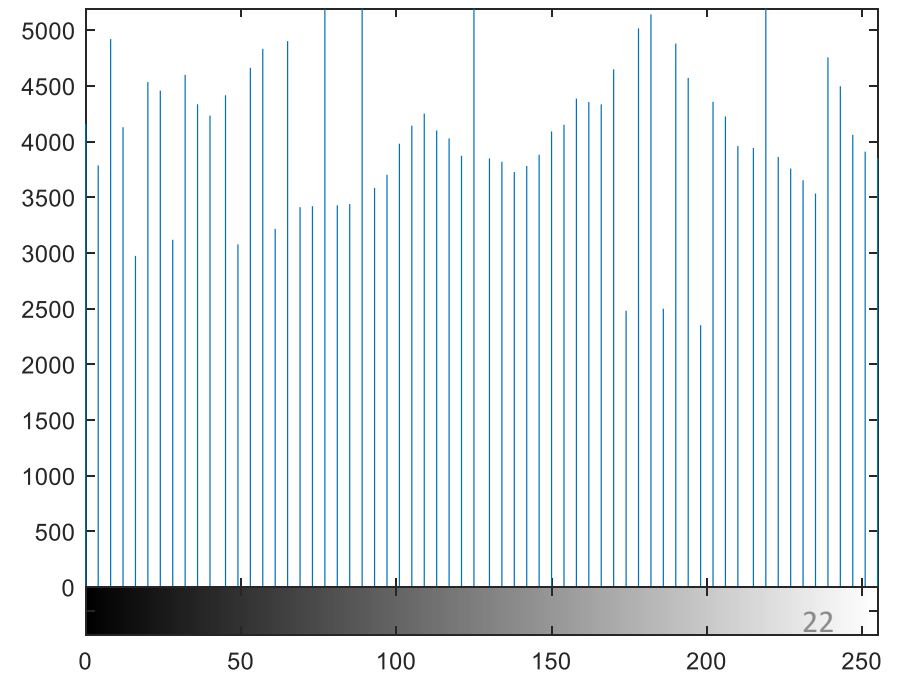
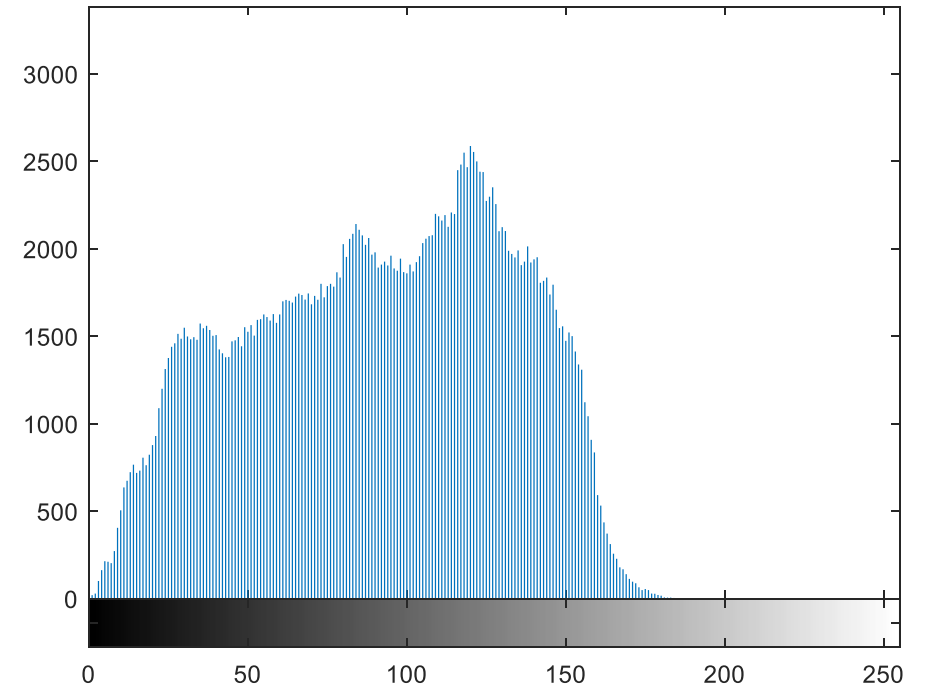
Histogram hasil perataan:

3	6	6	3
8	3	8	6
6	3	6	9
3	8	3	8

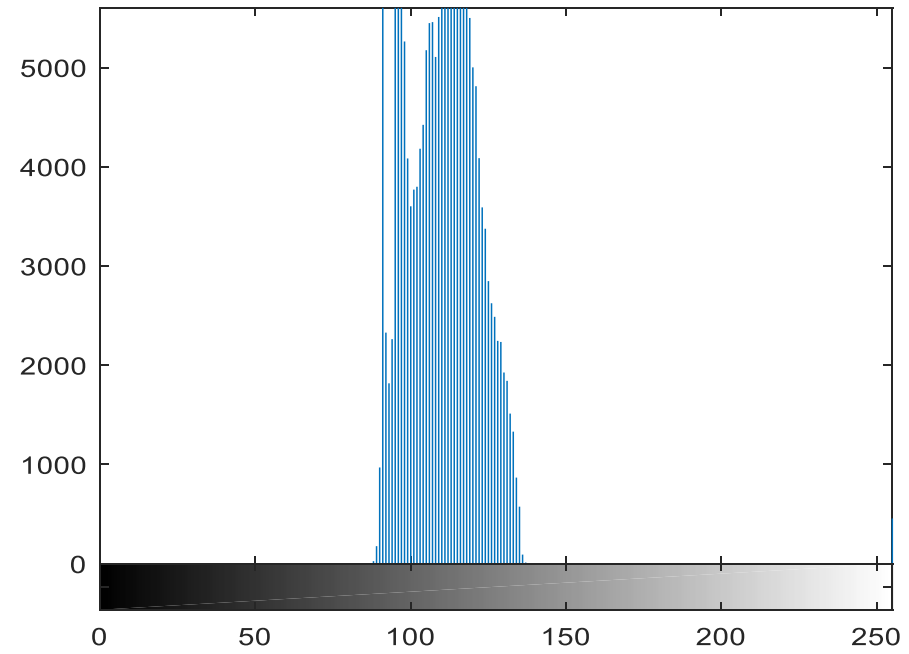
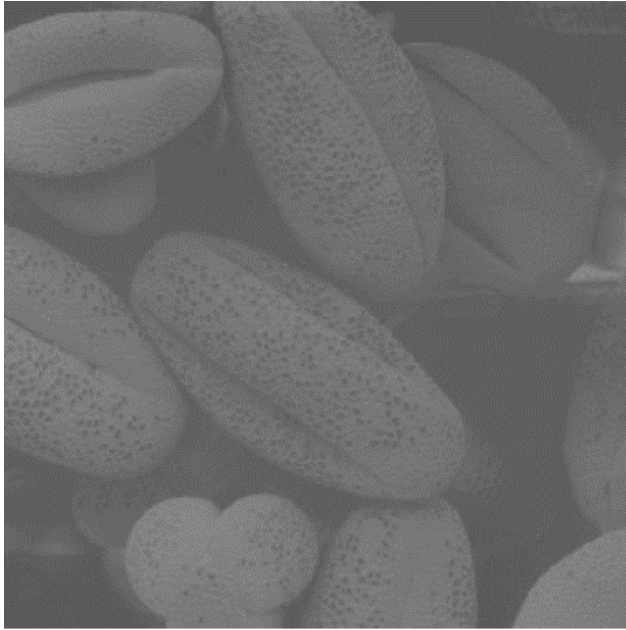
Output image
Gray scale = [0,9]



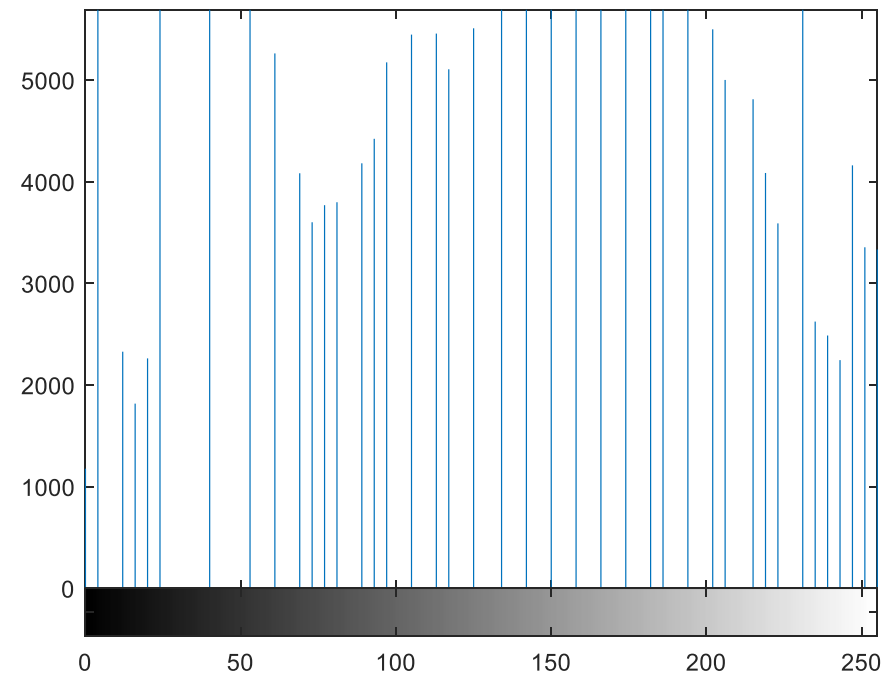
```
>> imread('zelda.bmp');  
>> imshow(I);  
>> figure; imhist(I)  
>> J = histeq(I);  
>> figure; imshow(J);  
>> figure; imhist(J);
```



Sebelum

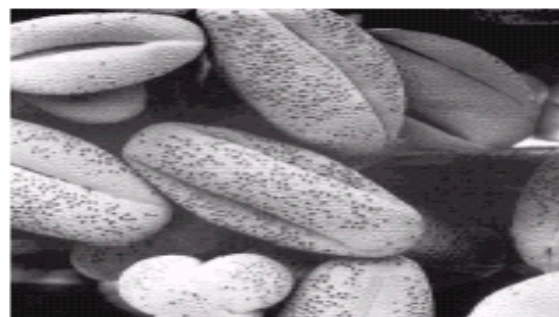
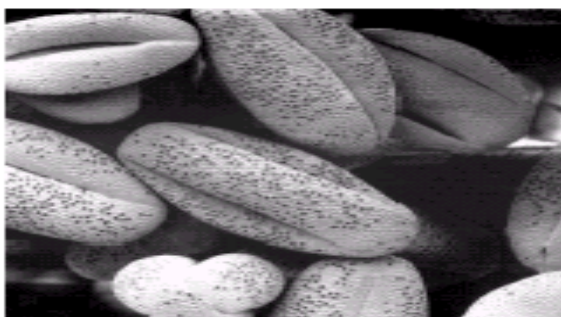
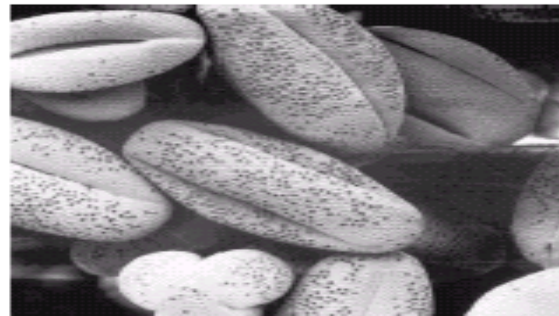
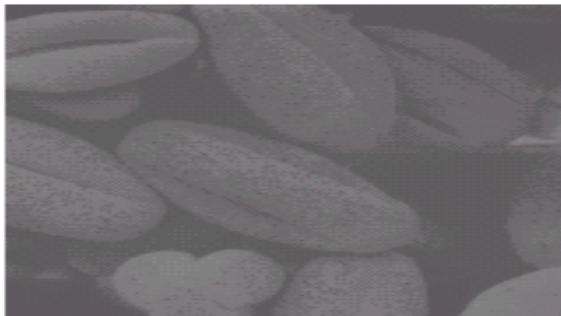
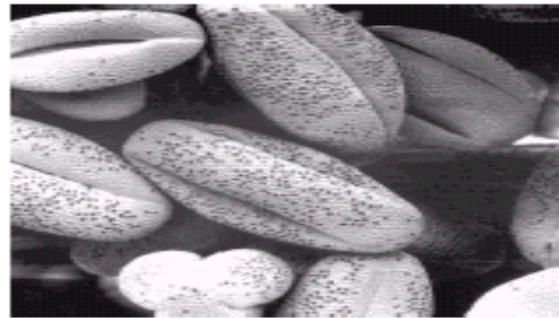
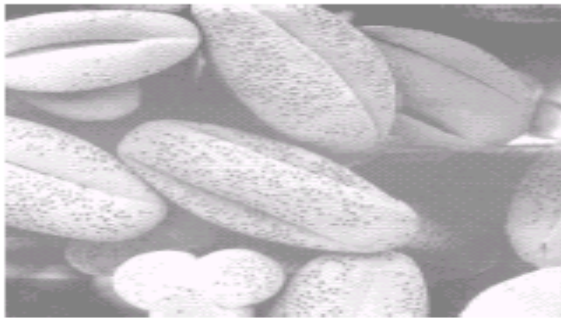
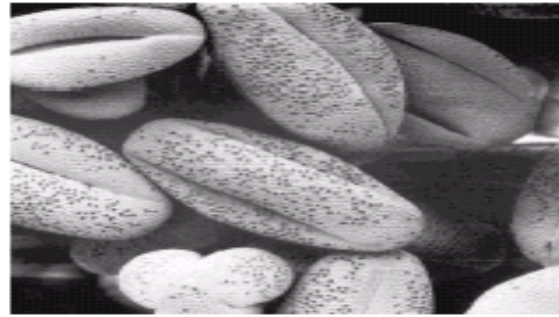
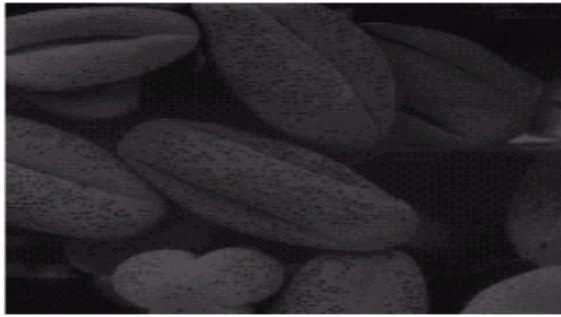


Sesudah

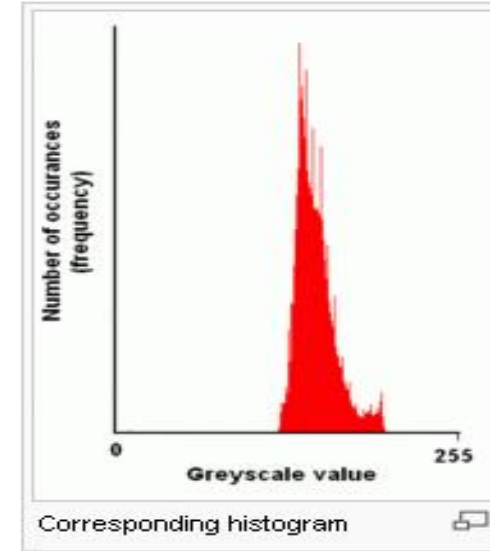


Sebelum

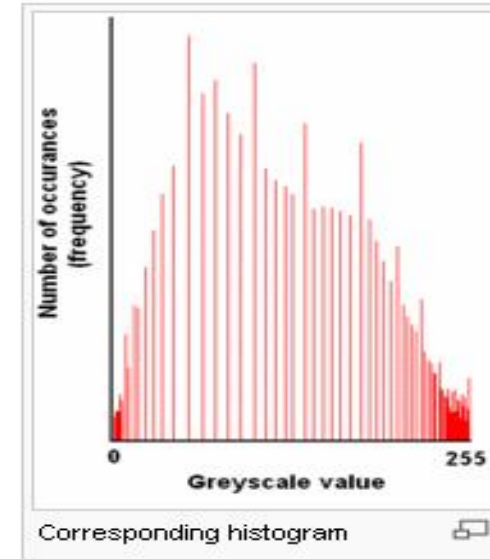
Sesudah



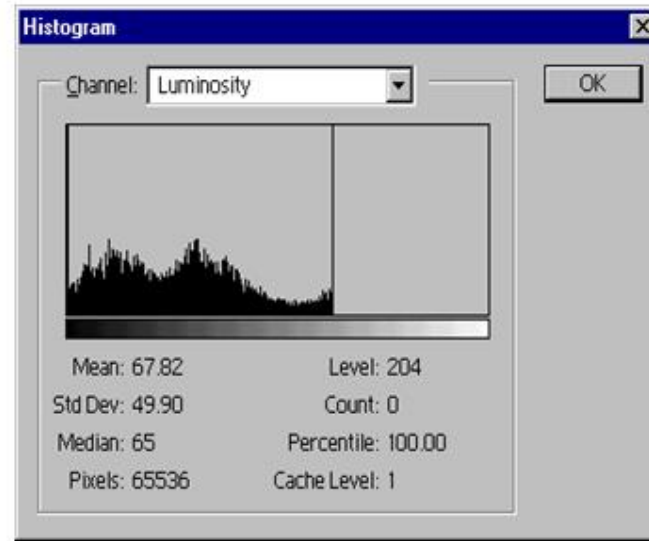
Sebelum



Sesudah

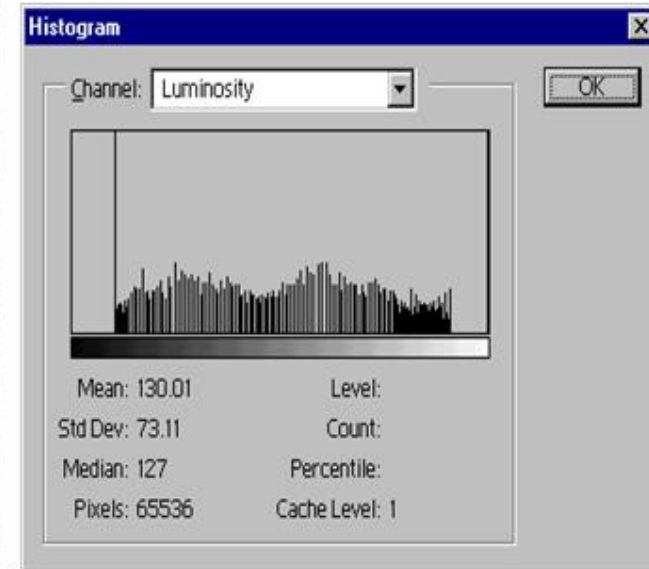


Sebelum

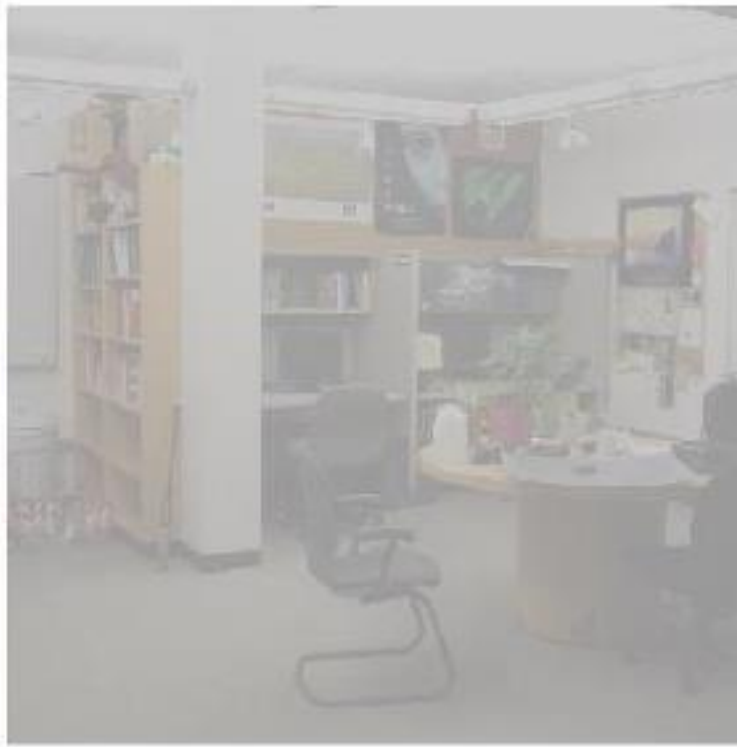


(a) Kiri: citra anjing *collie* yang terlalu gelap; Kanan: histogramnya

Sesudah



(b) Kiri: citra anjing *collie* setelah perataan histogram; kanan: histogramnya

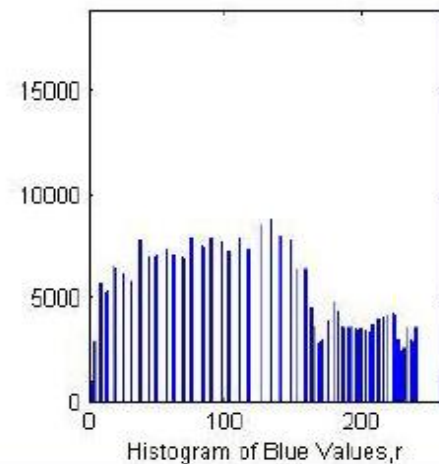
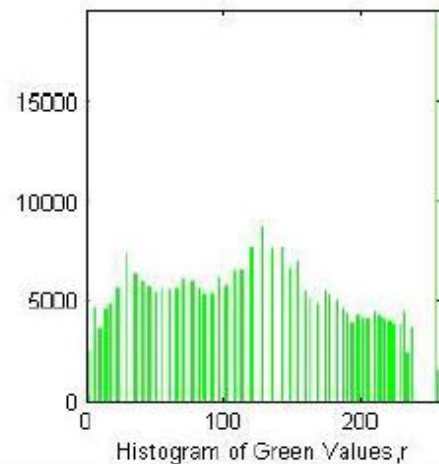
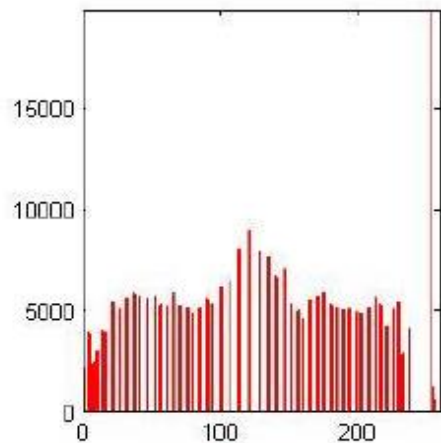


Original Image



Enhanced Image by Histogram Equalization

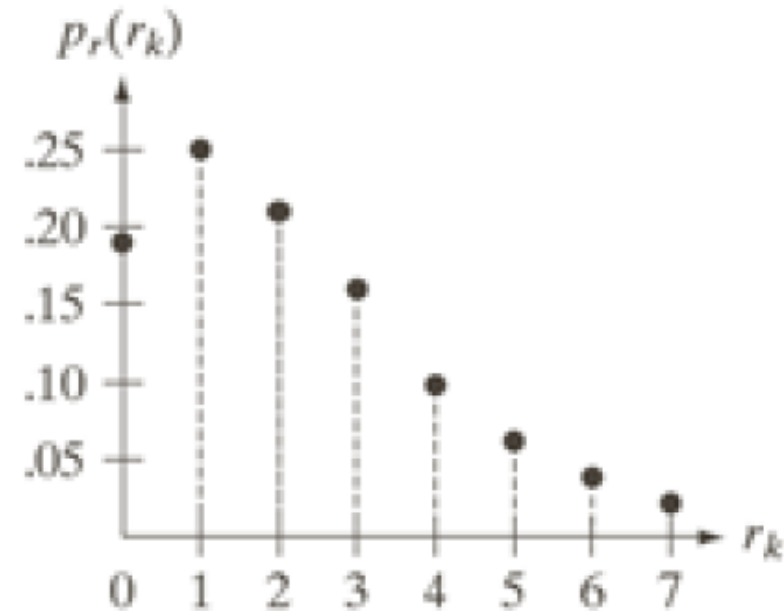
Histogram setelah perataan:



Sumber gambar: Ehsan Khoramshahi,
Image enhancement in spatial domain

Contoh: Misalkan terdapat citra yang berukuran 64×64 dengan jumlah derajat keabuan (L) = 8 dan jumlah seluruh *pixel* (n) = $64 \times 64 = 4096$.

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02



Perhitungan perataan histogram adalah sbb:

$$s_0 = T(r_0) = \sum_{j=0}^0 P_r(r_j) = P_r(r_0) = 0.19$$

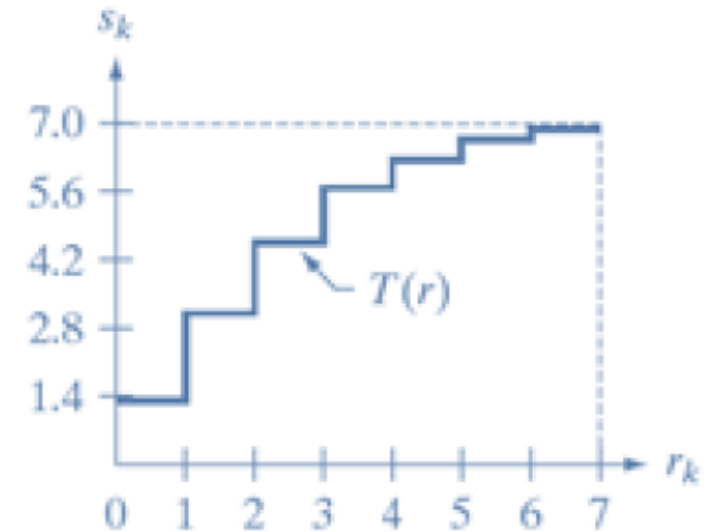
$$s_1 = T(r_1) = \sum_{j=0}^1 P_r(r_j) = P_r(r_0) + P_r(r_1) = 0.19 + 0.25 = 0.44$$

$$s_2 = T(r_2) = \sum_{j=0}^2 P_r(r_j) = P_r(r_0) + P_r(r_1) + P_r(r_2) = 0.19 + 0.25 + 0.21 = 0.65$$

dan seterusnya, diperoleh:

$$\begin{array}{ll} s_3 = 0.81 & s_6 = 0.98 \\ s_4 = 0.89 & s_7 = 1.00 \\ s_5 = 0.95 & \end{array}$$

$$\begin{array}{l} s_0 = 0.19 \times 7 = 1.33 \approx 1 \\ s_1 = 0.44 \times 7 = 3.08 \approx 3 \\ s_2 = 0.65 \times 7 = 4.55 \approx 5 \\ s_3 = 0.81 \times 7 = 5.67 \approx 6 \\ s_4 = 0.89 \times 7 = 6.23 \approx 6 \\ s_5 = 0.95 \times 7 = 6.65 \approx 7 \\ s_6 = 0.98 \times 7 = 6.86 \approx 7 \\ s_7 = 1.00 \times 7 = 7.0 = 7 \end{array}$$



Tabel awal:

r_k	n_k	$P_r(r_k) = n_k/n$
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

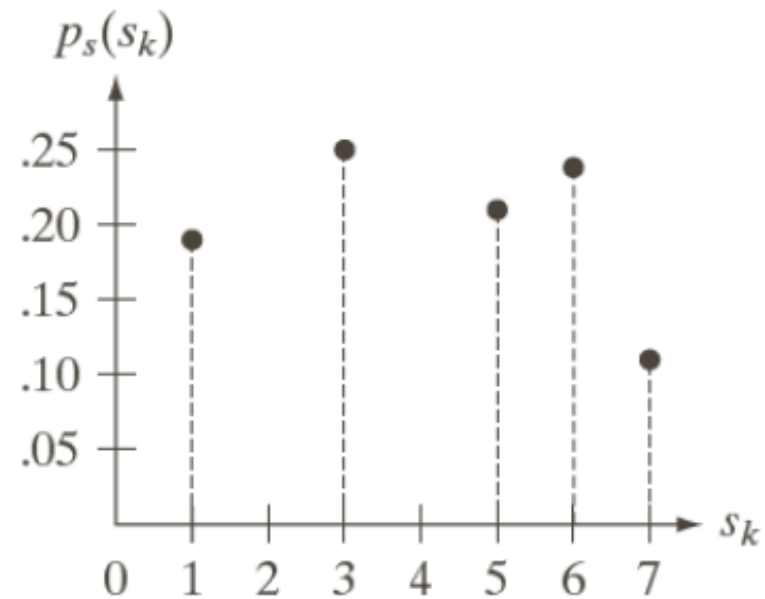
Hasil transformasinya :

k	r_k	S_k
0	0	1
1	1	3
2	2	5
3	3	6
4	4	6
5	5	7
6	6	7
7	1	7

- $r_0 = 0$ dipetakan ke $s_0 = 1$, maka terdapat 790 *pixel* yang memiliki nilai keabuan 1.
- $r_1 = 1$ dipetakan ke $s_1 = 3$, maka $s_1 = 3$ memiliki 1023 *pixel*,
- $r_2 = 2$ dipetakan ke $s_2 = 5$, maka $s_2 = 5$ memiliki 850 *pixel*.
- r_3 dan r_4 dipetakan ke nilai yang sama, $s_3 = 6$, maka jumlah *pixel* bernilai 6 = 656 + 329 = 985.
- r_5 , r_6 dan r_7 dipetakan ke nilai yang sama, $s_4 = s_4 = s_4 = 7$, maka jumlah *pixel* bernilai 7 = 245 + 122 + 81 = 448.

s_k	n_k	$P_s(s_k) = n_k/n$
1	790	0.19
3	1023	0.25
5	850	0.21
6	$656 + 329 = 958$	0.23
7	$245 + 122 + 81 = 448$	0.11

Histogram hasil perataan:



```

void PerataanHistogram(citra Image, int N, int M)
/* Mengubah citra Image yang berukuran N x M dengan melakukan perataan
histogram (histogram equalization).
*/
{
    int i, j;
    float sum, float Hist[256];
    int HistEq[256];          /* histogram hasil perataan */

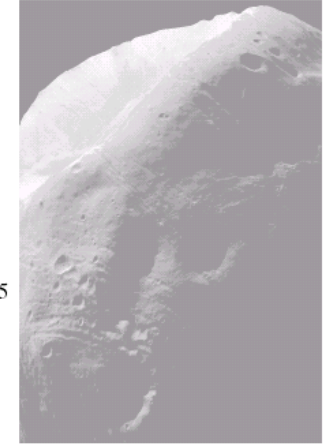
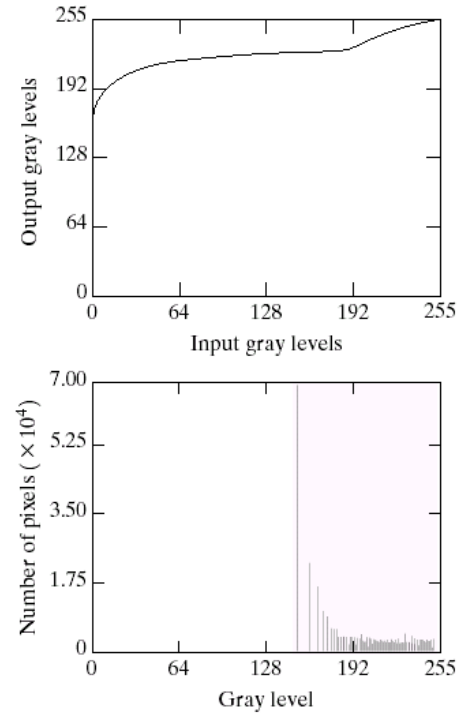
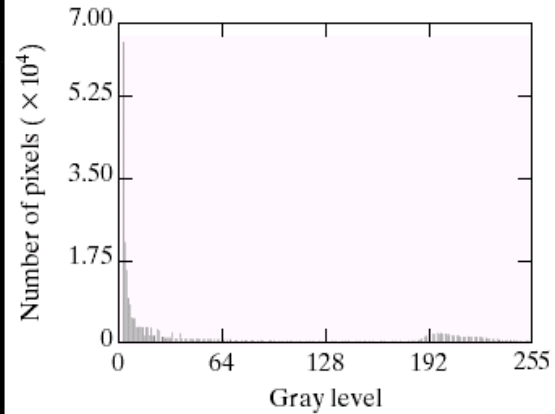
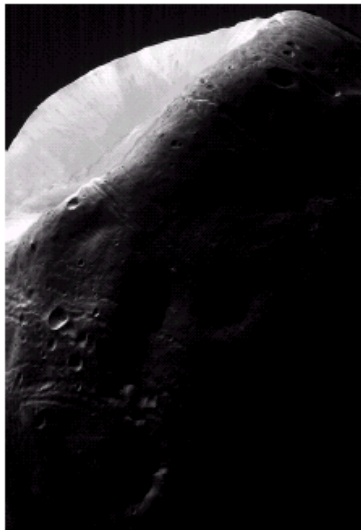
    histogram(Image,N,M,Hist); /* hitung histogram citra */
    for(i=0;i<256;i++)
    {
        sum=0.0;
        for (j=0;j<=i;j++)
            sum=sum+Hist[j];
        HistEq[i]=floor(255*sum);
    }
    /* update citra sesuai histogram hasil perataan */
    for(i=0;i<=N-1;i++)
        for(j=0;j<=M-1;j++)
            Image[i][j]=HistEq[Image[i][j]];
}

```


- Meskipun perataan histogram bertujuan menyebarkan secara merata nilai-nilai derajat keabuan, tetapi seringkali histogram hasil perataan tidak benar-benar tersebar secara merata (misalnya pada contoh di atas). Alasannya adalah :
 1. Derajat keabuan terbatas jumlahnya. Nilai intensitas baru hasil perataan merupakan pembulatan ke derajat keabuan terdekat.
 2. Jumlah *pixel* yang digunakan sangat terbatas.
- Agar hasil perataan benar-benar seragam sebarannya, maka citra yang diolah haruslah dalam bentuk malar (*continue*), yang dalam praktek ini jelas tidak mungkin.

2. Spesifikasi Histogram

- Perataan histogram memetakan histogram citra semula menjadi histogram yang seragam.
- Bila histogram yang diinginkan tidak seragam, maka cara ini tidak dapat digunakan.



Perataan histogram memberikan efek terlalu terang pada citra hasil
(Citra permukaan bulan)

- Metode **spesifikasi histogram** (*histogram specification*) atau **pencocokan histogram** (*histogram matching*) memberikan cara menghasilkan histogram yang ditentukan oleh pengguna.
- Cara pembentukan histogramnya memanfaatkan sifat pada perataan histogram.
- Bila fungsi transformasi pada perataan histogram menghasilkan histogram semula menjadi histogram yang seragam, maka fungsi balikkannya (*inverse*) memetakan histogram yang seragam menjadi histogram semula.
- Sifat ini dapat dimanfaatkan untuk mengubah histogram citra menjadi histogram lain yang tidak seragam.

- Misalkan $P_r(r)$ dan $P_z(z)$ masing-masing adalah histogram citra semula dan histogram yang diinginkan.
- Fungsi transformasi T mula-mula memetakan intensitas citra semula menjadi histogram yang seragam dengan cara perataan histogram,

$$s = T(r) = \int_0^r P_r(w)dw$$

- Jika histogram yang diinginkan sudah dispesifikasikan, kita dapat melakukan perataan histogram pula dengan fungsi transformasi G :

$$v = G(z) = \int_0^z P_z(w)dw$$

- Balikan (*invers*) dari fungsi G ,

$$z = G^{-1}(v)$$

akan menghasilkan histogram yang diinginkan kembali.

- Dengan mengganti v dengan s pada persamaan yang terakhir,

$$z \approx G^{-1}(s)$$

- maka kita dapat memperoleh nilai intensitas yang diinginkan.
- Hasil yang diperoleh merupakan hampiran karena kita mencoba menemukan nilai s yang transformasinya mendekati nilai z .

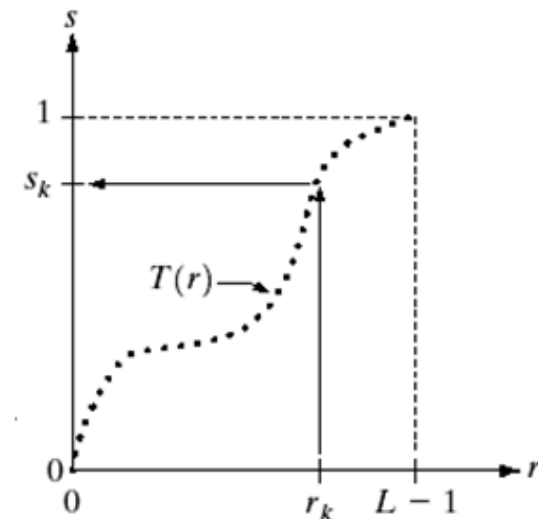
Algoritma spesifikasi histogram adalah sebagai berikut:

1. Misalkan $P_r(r)$ adalah histogram citra semula. Lakukan perataan histogram terhadap citra semula dengan fungsi transformasi T ,

$$s = T(r) = \int_0^r P_r(w)dw$$

Dalam bentuk diskrit, nilai-nilai s diperoleh dengan persamaan berikut:

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

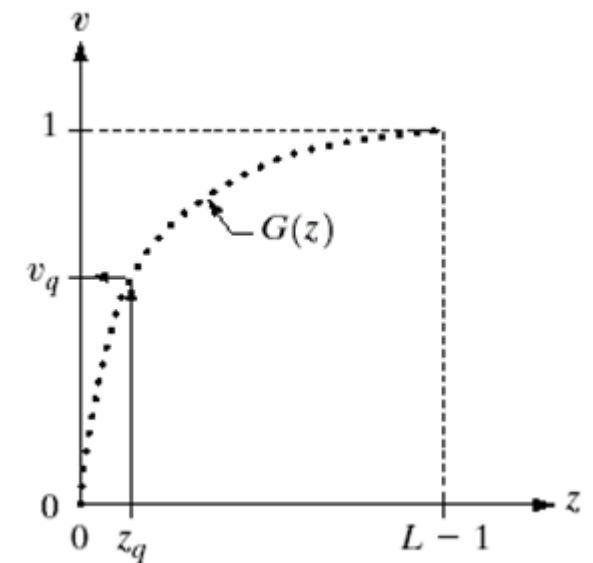


2. Tentukan histogram yang diinginkan, misalkan $P_z(z)$ adalah histogram yang diinginkan. Lakukan perataan histogram dengan fungsi transformasi G ,

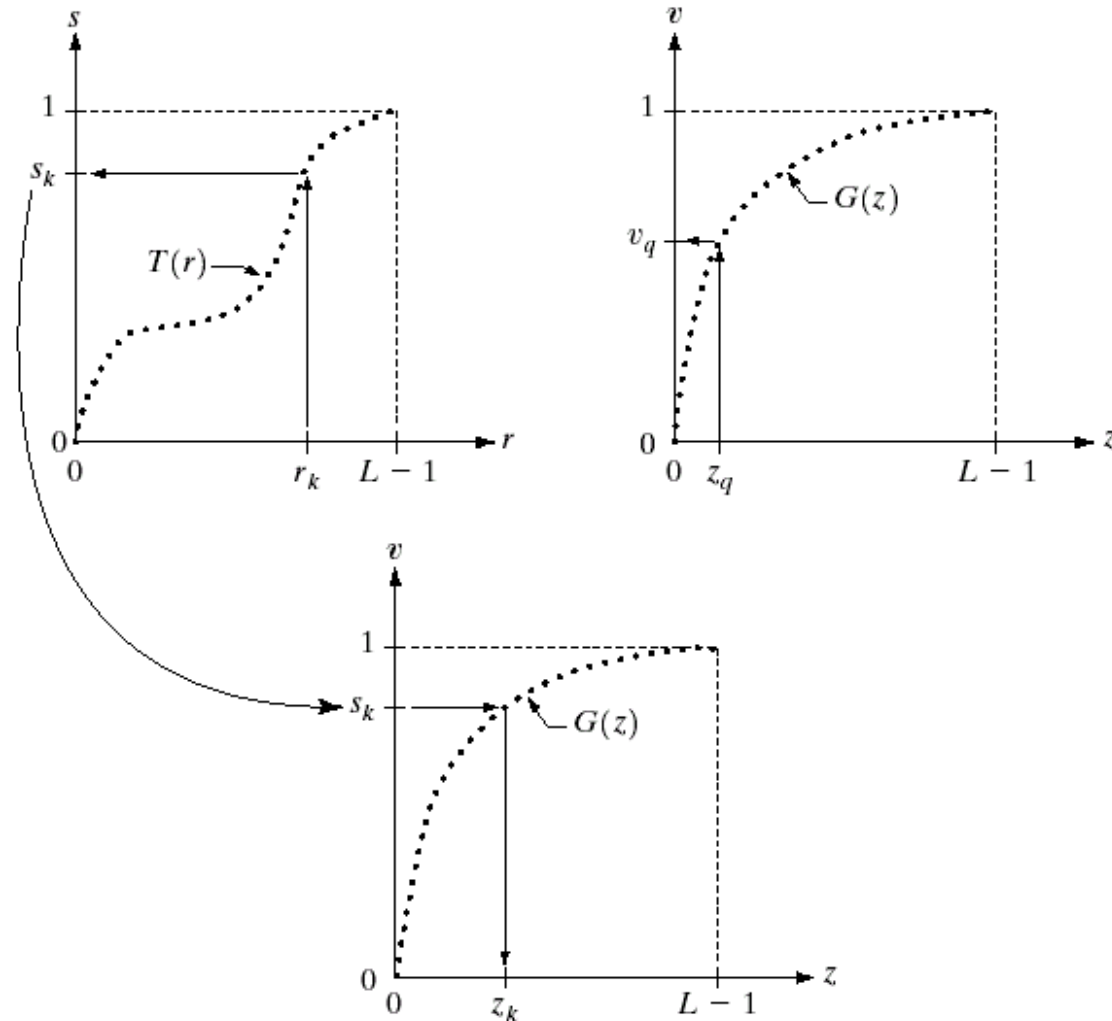
$$v = G(z) = \int_0^z P_z(w)dw$$

Dalam bentuk diskrit, nilai-nilai v diperoleh dengan persamaan berikut:

$$v_k = G(z_k) = \sum_{j=0}^k \frac{n_j}{n} = \sum_{j=0}^k P_z(z_j)$$



3. Terapkan fungsi transformasi balikan, $z = G^{-1}(s)$ terhadap histogram hasil langkah 1. Caranya adalah dengan mencari nilai-nilai s yang memberi nilai z terdekat.



- Dengan kata lain, histogram nilai-nilai intensitas pada citra semula dipetakan menjadi intensitas z pada citra yang diinginkan dengan fungsi

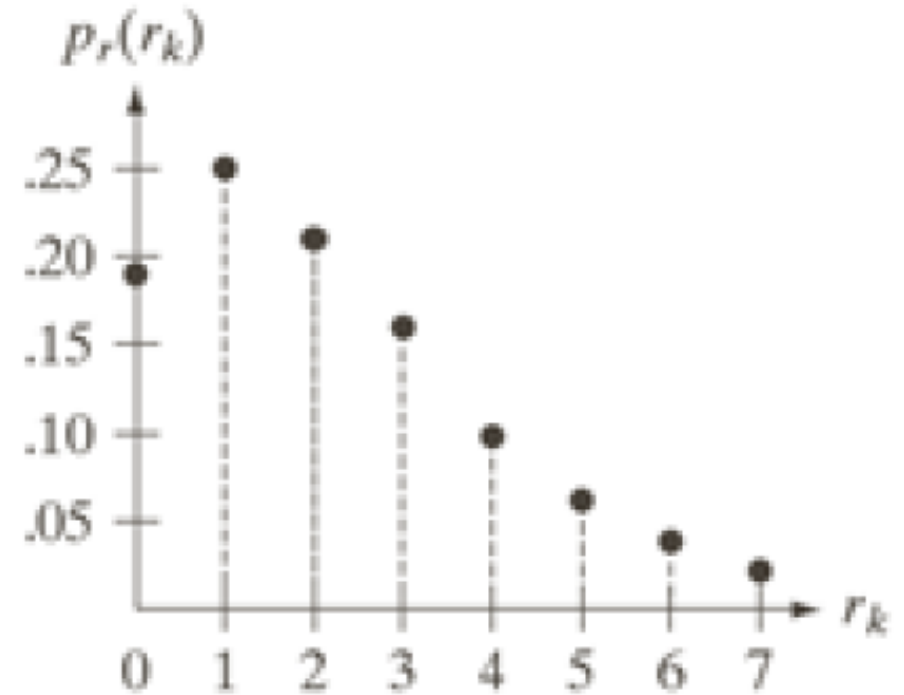
$$z = G^{-1}[T(r)]$$

- Ketiga langkah di dalam algoritma spesifikasi histogram di atas digambarkan dalam bagan pada gambar berikut:



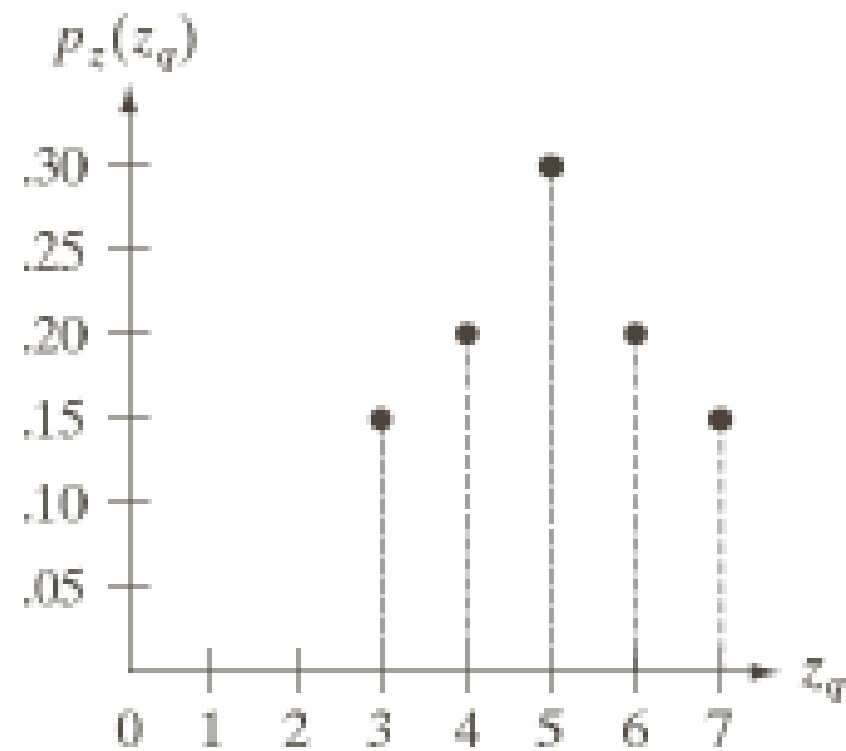
Contoh: Tinjau kembali citra yang berukuran 64×64 dengan jumlah derajat keabuan (L) = 8 dan jumlah seluruh *pixel* (n) = $64 \times 64 = 4096$.

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02



Histogram yang diinginkan:

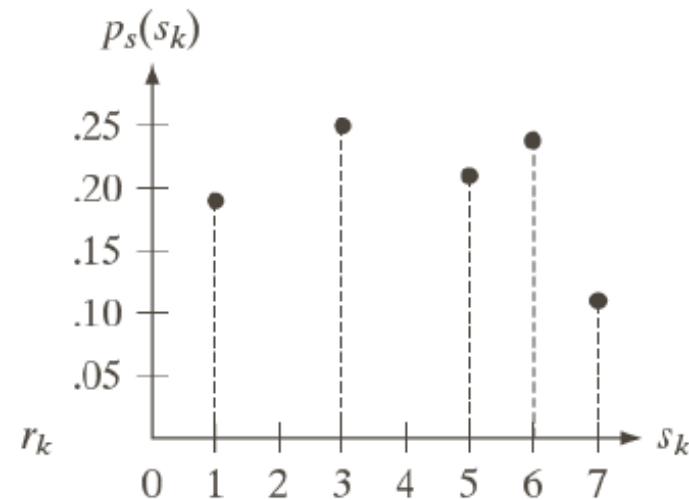
z_q	$P_z(z_q)$
$Z_0=0$	0.00
$Z_1=1$	0.00
$Z_2=2$	0.00
$Z_3=3$	0.15
$Z_4=4$	0.20
$Z_5=5$	0.30
$Z_6=6$	0.20
$Z_7=7$	0.15



Step 1: Lakukan perataan histogram terhadap citra semula

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

s_0	1.33	1
s_1	3.08	3
s_2	4.55	5
s_3	5.67	6
s_4	6.23	6
s_5	6.65	7
s_6	6.86	7
s_7	7.00	7



s_k	n_k	$P_s(s_k) = n_k/n$
1	790	0.19
3	1023	0.25
5	850	0.21
6	$656 + 329 = 958$	0.23
7	$245 + 122 + 81 = 448$	0.11

Step 2: Lakukan perataan terhadap histogram yang diinginkan, $P_z(z)$

$$v_k = G(z_k) = \sum_{j=0}^k \frac{n_j}{n} = \sum_{j=0}^k P_z(z_j)$$

• Hasilnya adalah sbb:

$$v_0 = G(z_0) = 0.00 \times 7 = 0$$

$$v_1 = G(z_1) = 0.00 \times 7 = 0$$

$$v_2 = G(z_2) = 0.00 \times 7 = 0$$

$$v_3 = G(z_3) = 0.15 \times 7 = 1.05$$

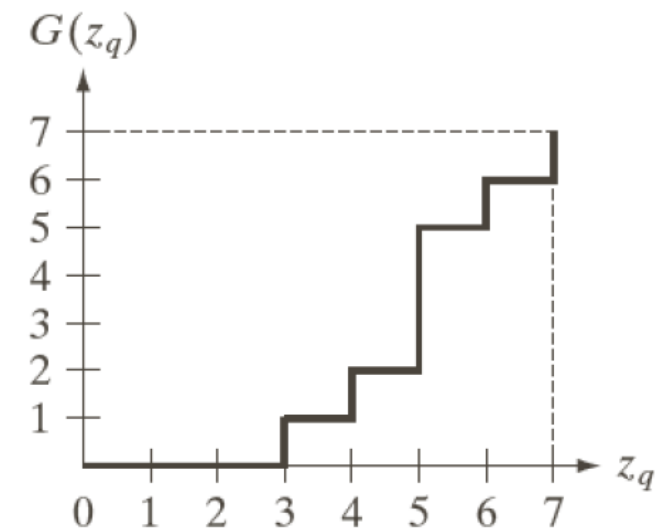
$$v_4 = G(z_4) = 0.35 \times 7 = 2.45$$

$$v_5 = G(z_5) = 0.65 \times 7 = 4.55$$

$$v_6 = G(z_6) = 0.85 \times 7 = 5.95$$

$$v_7 = G(z_7) = 1.00 \times 7 = 7$$

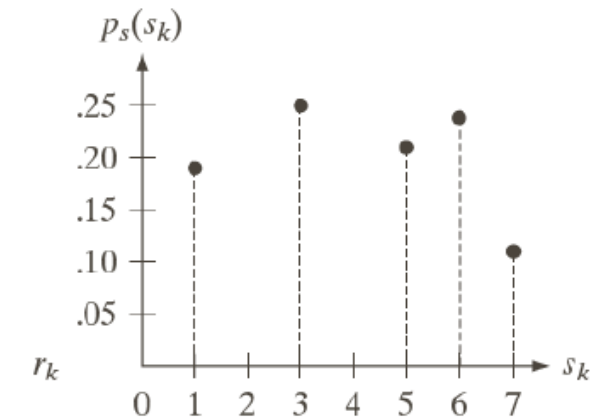
$G(z_0)$	0.00	0
$G(z_1)$	0.00	0
$G(z_2)$	0.00	0
$G(z_3)$	1.05	1
$G(z_4)$	2.45	2
$G(z_5)$	4.55	5
$G(z_6)$	5.95	6
$G(z_7)$	7.00	7



Step 3: Gunakan transformasi $z = G^{-1}(s)$ untuk memperoleh nilai z dari nilai s hasil perataan histogram.

- $s_0 = 1$ paling dekat dengan $1.05 = G(z_3)$, jadi $G^{-1}(1) = z_3 = 1$
- $s_1 = 3$ paling dekat dengan $2.45 = G(z_4)$, jadi $G^{-1}(3) = z_4 = 4$
- $s_2 = 5$ paling dekat dengan $4.55 = G(z_5)$, jadi $G^{-1}(5) = z_5 = 5$
- $s_3 = s_4 = 6$ paling dekat dengan $5.95 = G(z_6)$, jadi $G^{-1}(6) = z_6 = 6$
- $s_5 = s_6 = s_7 = 7$ paling dekat dengan $7.00 = G(z_7)$, jadi $G^{-1}(7) = z_7 = 7$

s_0	1.33	1
s_1	3.08	3
s_2	4.55	5
s_3	5.67	6
s_4	6.23	6
s_5	6.65	7
s_6	6.86	7
s_7	7.00	7



$G(z_0)$	0.00	0
$G(z_1)$	0.00	0
$G(z_2)$	0.00	0
$G(z_3)$	1.05	1
$G(z_4)$	2.45	2
$G(z_5)$	4.55	5
$G(z_6)$	5.95	6
$G(z_7)$	7.00	7

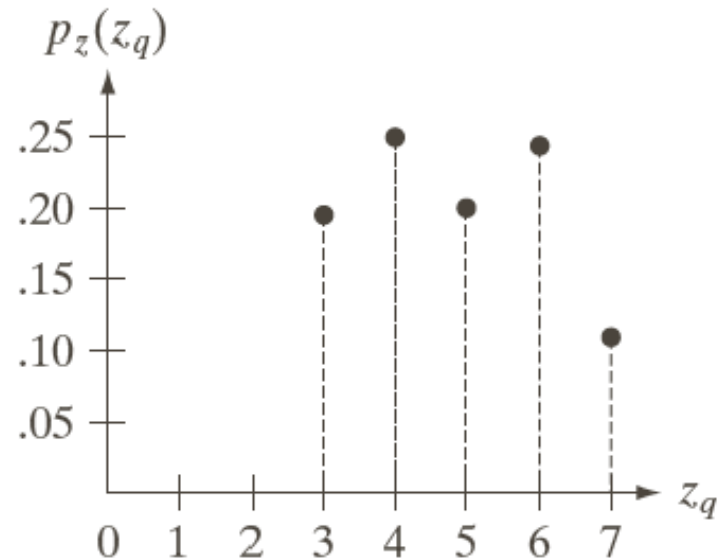
Hasil pemetaan:

s_k	z_q
1	3
3	4
5	5
6	6
7	7

Jumlah pixel:

s_k	n_k
1	790
3	1023
5	850
6	$656 + 329 = 958$
7	$245 + 122 + 81 = 448$

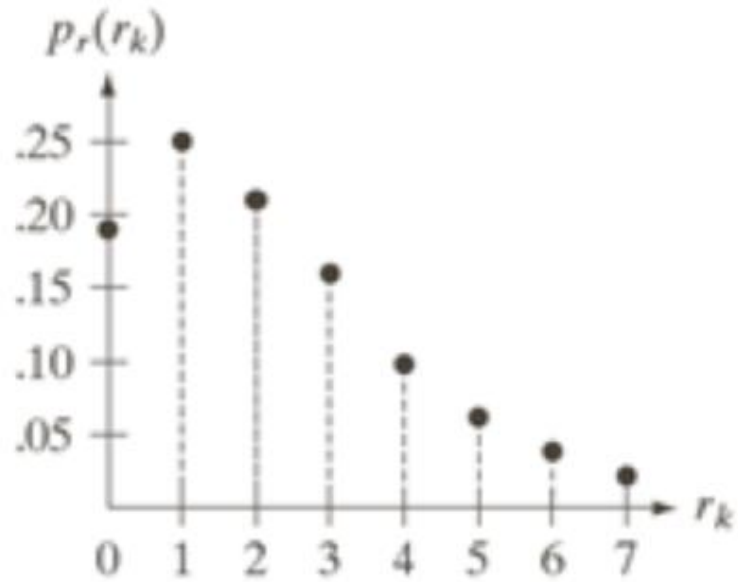
Histogram:



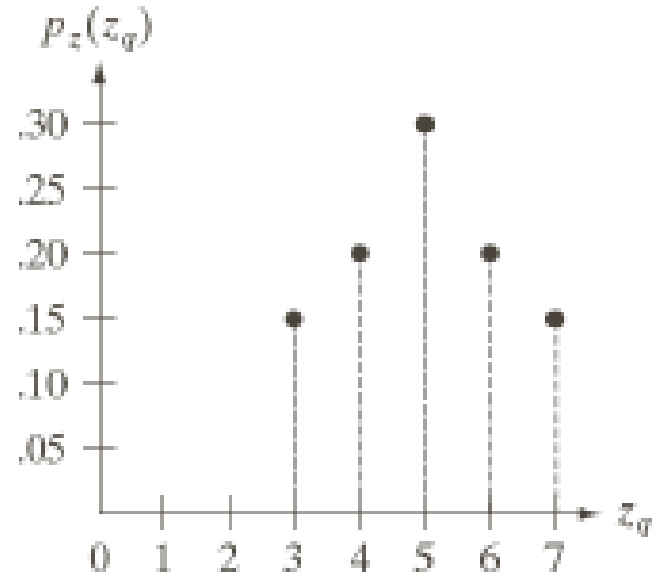
Jumlah pixel:

z_k	n_k	$P_z(z_k) = n_k/n$
0	0	0.00
1	0	0.00
2	0	0.00
3	790	0.19
4	1023	0.25
5	850	0.21
6	985	0.24
7	448	0.11

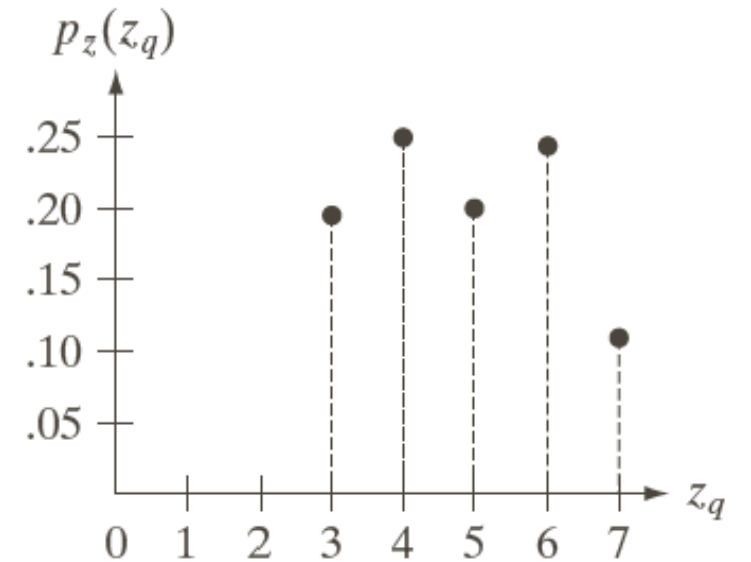
Perbandingan:



Histogram citra semula



Histogram citra yang diinginkan



Histogram citra hasil

```

void SpesifikasiHistogram(citra Image, int N, int M, float Spec[256])
/* Mengubah citra Image yang berukuran N x M berdasarkan histogram yang dispesifikasikan
oleh pengguna (Spec). */
{
    float sum, Hist[256];
    int i, j, minj, minval, HistEq[256], SpecEq[256], InvHist[256];

    /* lakukan perataan histogram terhadap citra semula */
    histogram(Image,N,M,Hist); /* hitung histogram citra */
    for(i=0;i<256;i++)
    {
        sum=0.0;
        for (j=0;j<=i;j++)
            sum=sum+Hist[j];
        HistEq[i]=floor(255*sum);
    }

    /* lakukan perataan histogram terhadap citra Spec */
    for(i=0;i<=255;i++)
    {
        sum=0.0;
        for (j=0;j<=i;j++)
            sum=sum+Spec[j];
        SpecEq[i]=floor(255*sum);
    }
}

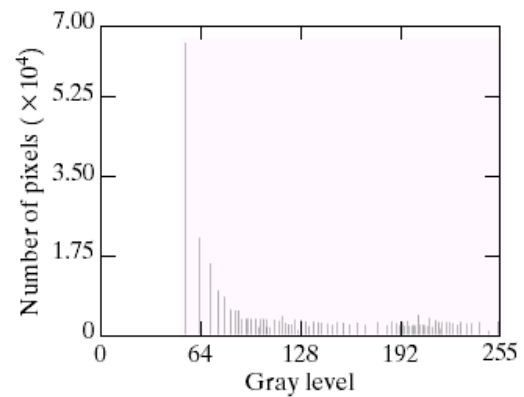
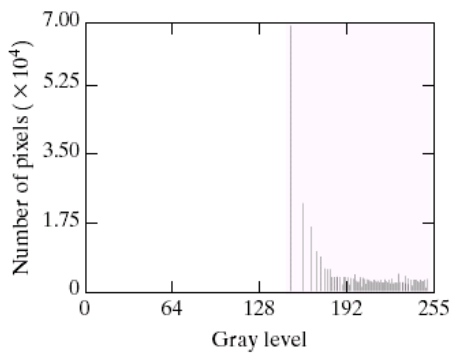
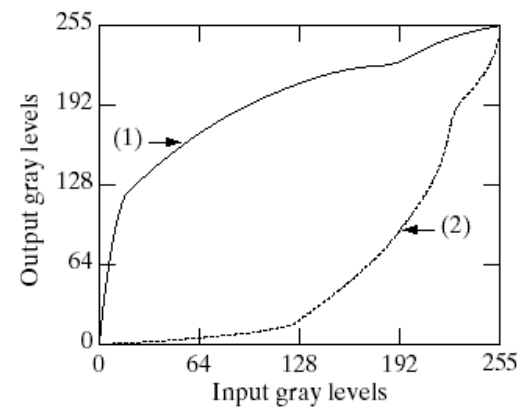
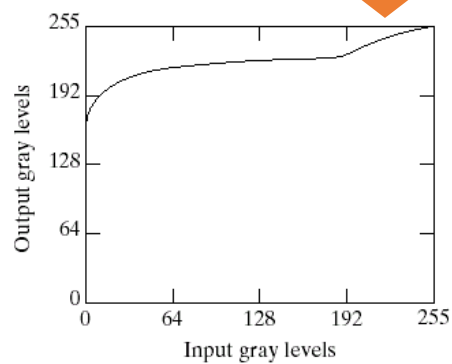
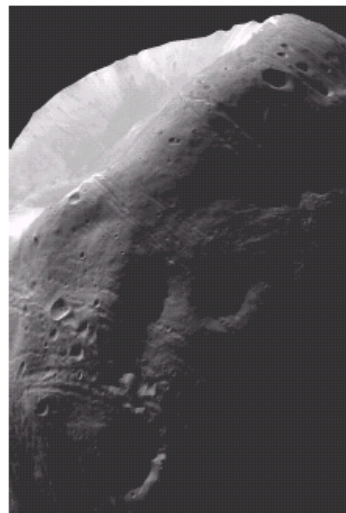
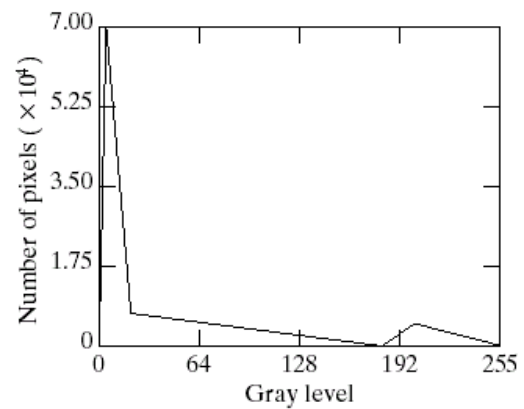
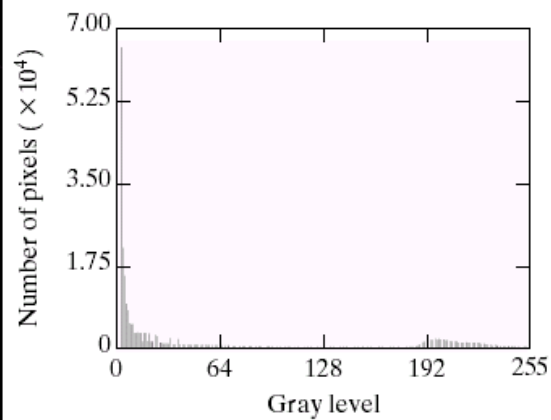
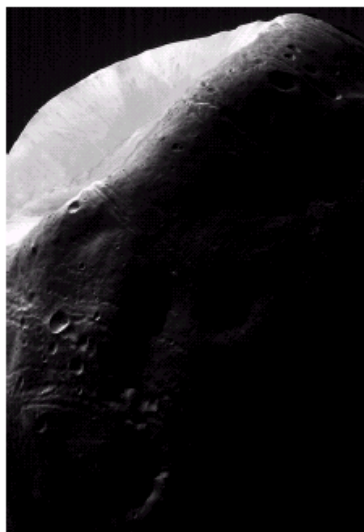
```

```

/* lakukan transformasi balikan */
for(i=0;i<=N-1;i++)
{
    minval=abs(HistEq[i] - SpecEq[0]);
    minj=0;
    for(j=0;j<=255;j++)
        if (abs(HistEq[i] - SpecEq[j]) < minval)
            {
                minval = abs(HistEq[i] - SpecEq[j]);
                minj=j;
            }
    InvHist[i]=minj;
}

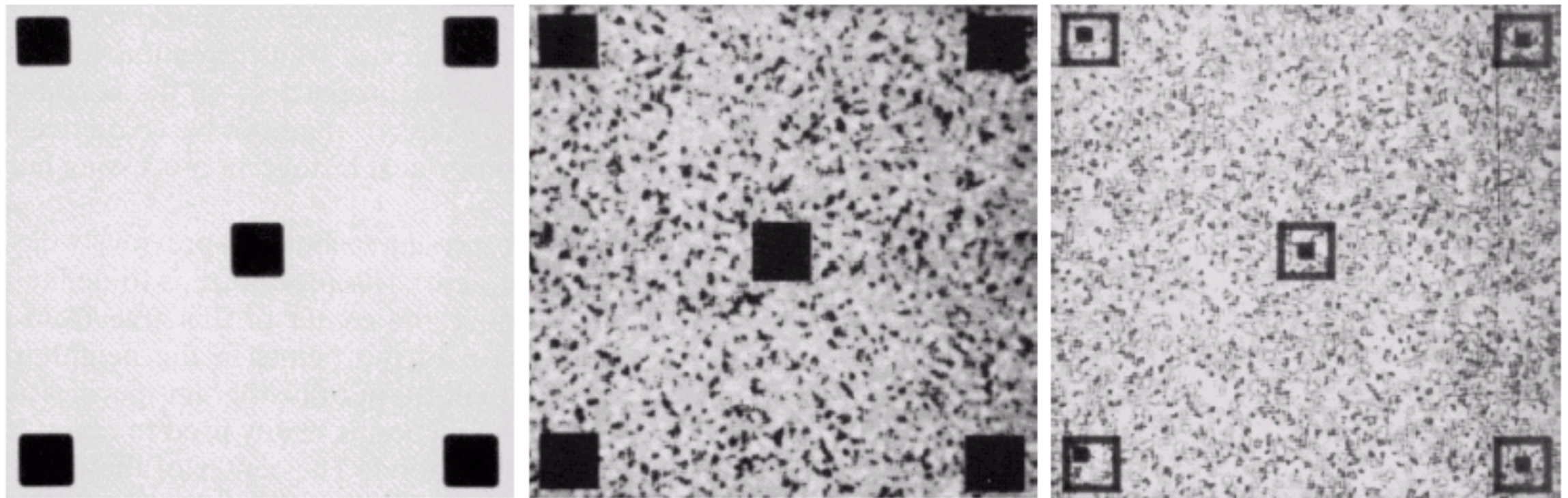
/* update citra setelah pembentukan histogram */
for(i=0;i<=N-1;i++)
    for(j=0;j<=M-1;j++)
        Image[i][j]=InvHist[Image[i][j]];
}

```



Perbaikan lokal dengan perataan histogram

- Perataan histogram tidak hanya dilakukan untuk keseluruhan *pixel* di dalam citra, tetapi juga hanya untuk area tertentu (ROI atau *region of interest*) di dalam citra.
- Ini artinya kita melakukan perbaikan secara lokal dengan metode perataan histogram.



a b c

FIGURE 3.23 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization using a 7×7 neighborhood about each pixel.

Perbaiki citra dengan operasi aritmetika/logika

- Dilakukan *pixel per pixel* antara dua atau lebih citra
- Operasi logika:
 - AND
 - OR
 - NOT
- Operasi aritmetika
 - Pengurangan
 - Penjumlahan (rata-rata)
 - Perkalian
 - Pembagian

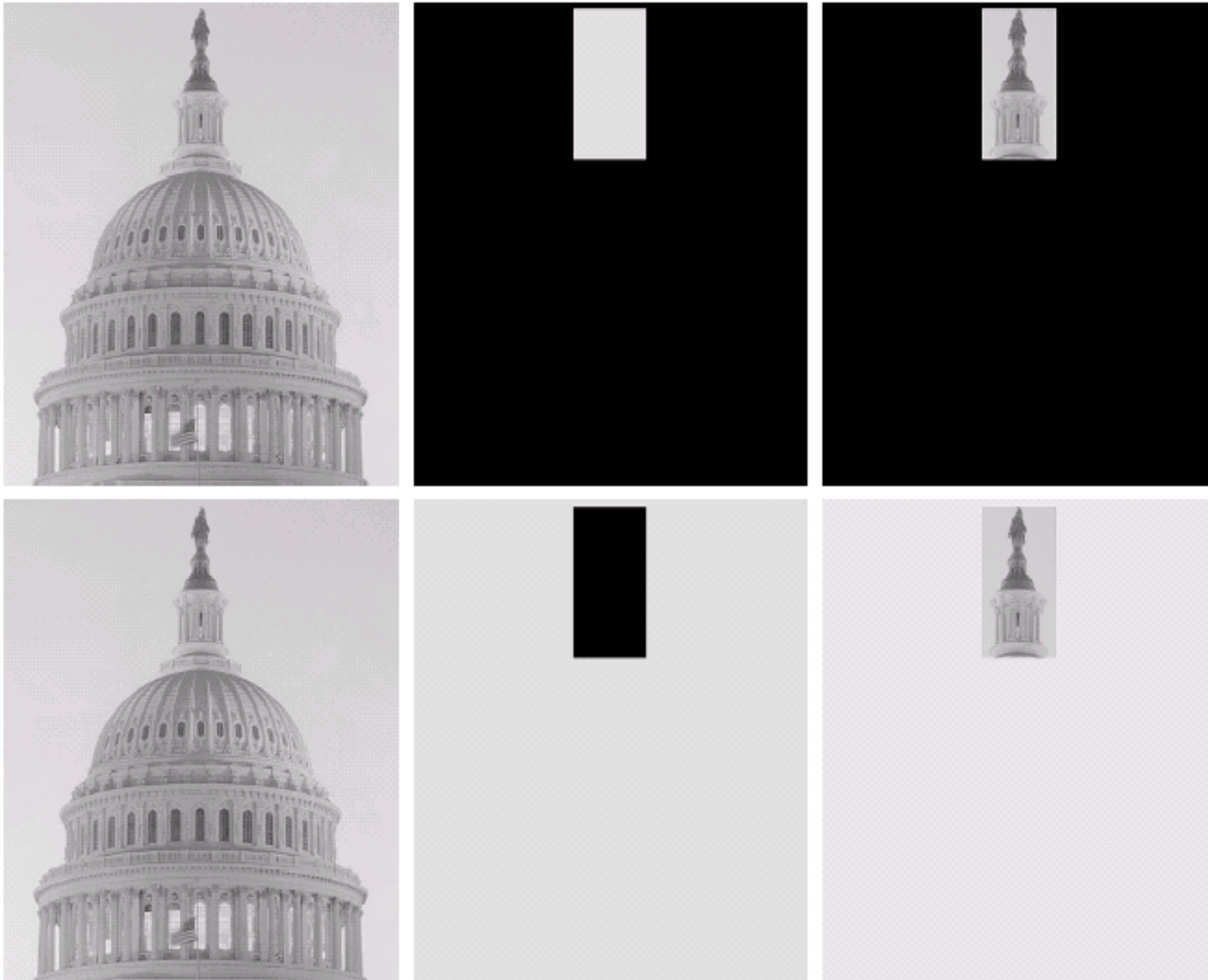
$$d(x,y) = f(x,y) - g(x,y)$$

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

$$p(x,y) = f(x,y) \times g(x,y)$$

$$v(x,y) = f(x,y) \div g(x,y)$$

1. Operasi AND dan OR

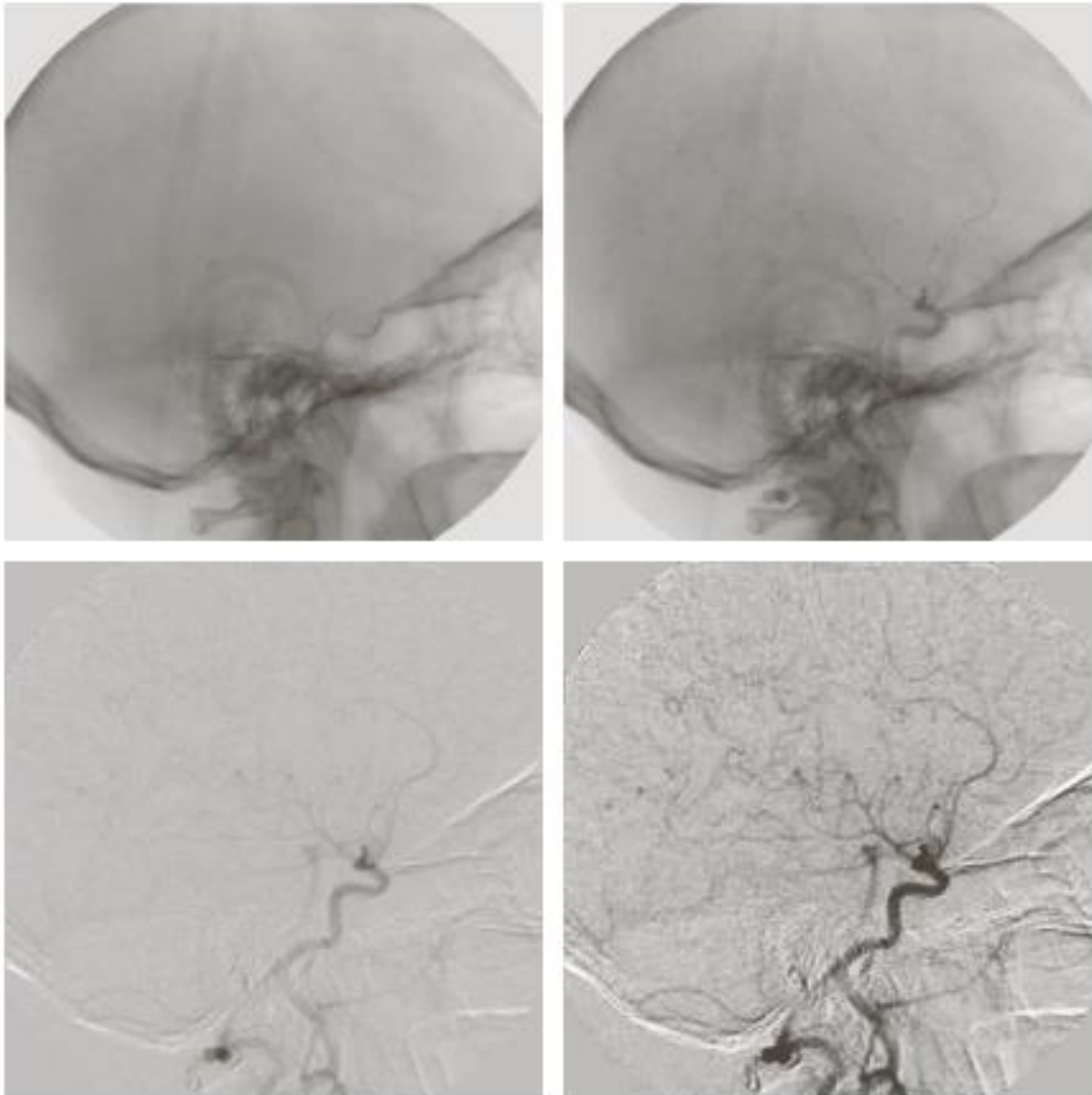


a	b	c
d	e	f

FIGURE 3.27

(a) Original image. (b) AND image mask. (c) Result of the AND operation on images (a) and (b). (d) Original image. (e) OR image mask. (f) Result of operation OR on images (d) and (e).

2. Operasi pengurangan



a b
c d

FIGURE 2.28

Digital subtraction angiography.

(a) Mask image.

(b) A live image.

(c) Difference

between (a) and

(b). (d) Enhanced

difference image.

(Figures (a) and

(b) courtesy of

The Image

Sciences Institute,

University

Medical Center,

Utrecht, The

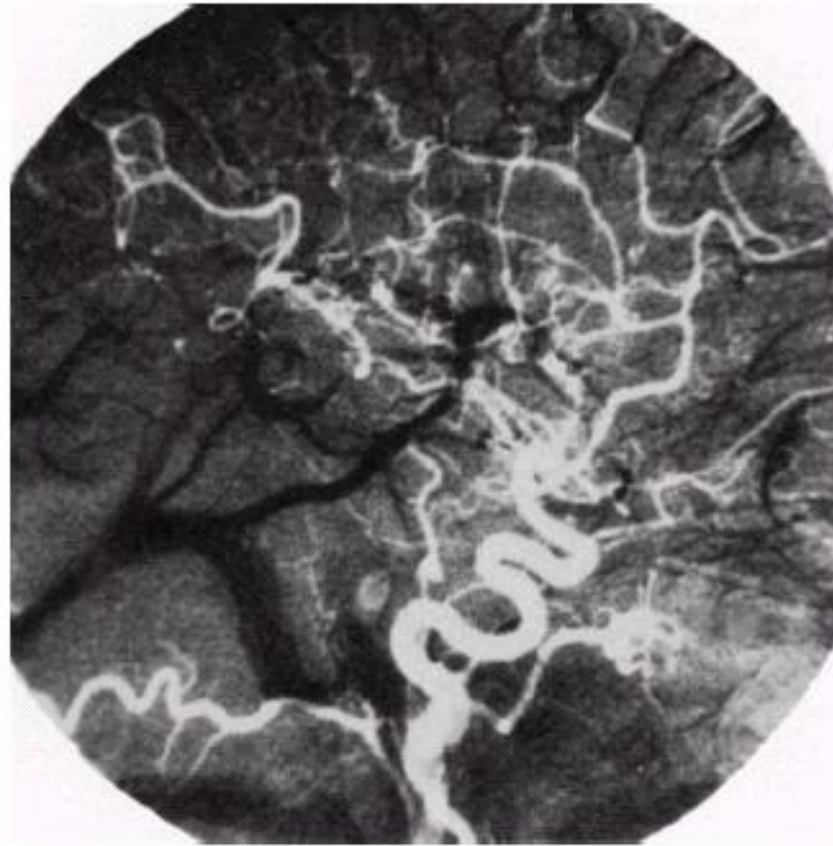
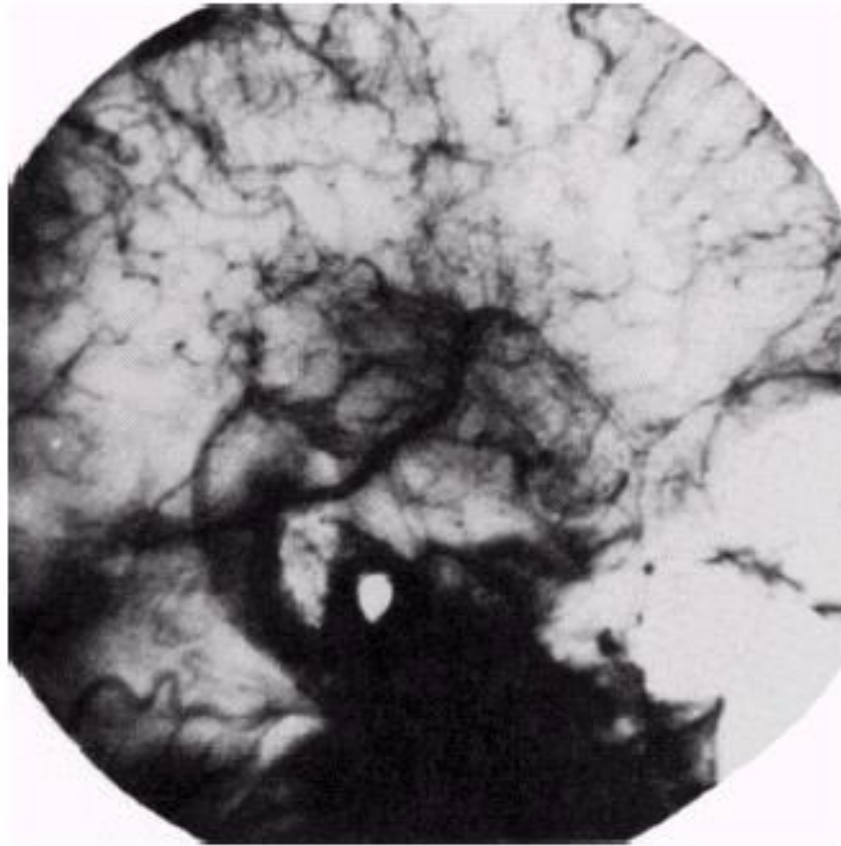
Netherlands.)

$$g(x,y) = f(x,y) - h(x,y)$$

Mask $h(x,y)$: an X-ray image of a region of a patient's body

Live images $f(x,y)$: X-ray images captured at TV rates after injection of the contrast medium

Sumber: Dr. Sanjeev Kumar, *Mathematical Imaging Techniques*, Department of Mathematics, IIT Roorkee



a b

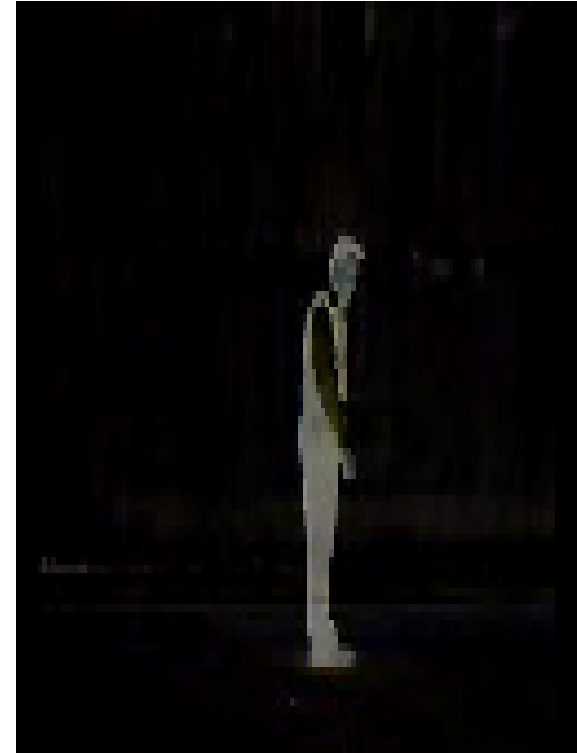
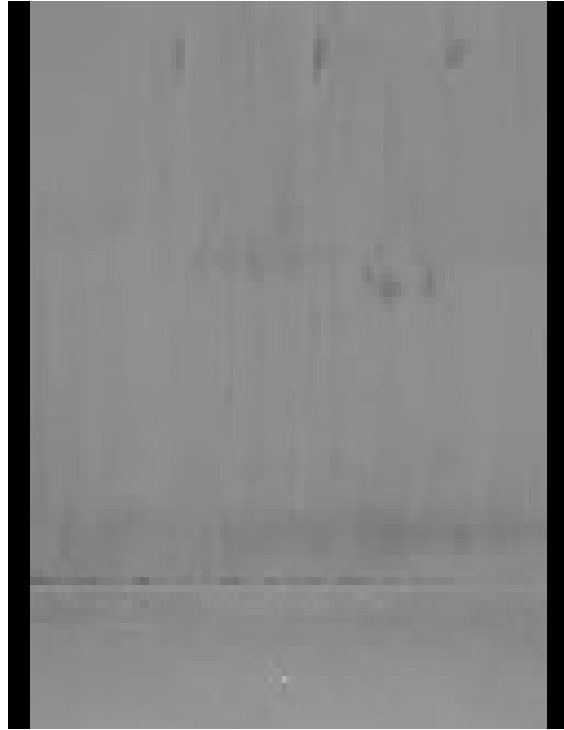
FIGURE 3.29

Enhancement by image subtraction. (a) Mask image. (b) An image (taken after injection of a contrast medium into the bloodstream) with mask subtracted out.



Background Removal Using Image Subtraction

Sumber: Ali Javed, Digital Image Processing, Chapter # 3, *Image Enhancement in Spatial Domain*



Background Removal Using Image Subtraction

3. Operasi penjumlahan (rata-rata)

Contoh dalam astronomi:

- Dalam astronomi, pencitraan di bawah tingkat cahaya yang sangat rendah sering menyebabkan gangguan sensor untuk menghasilkan gambar yang hampir tidak berguna untuk analisis.
- Dalam pengamatan astronomi, untuk mengurangi derau akibat sensor yang serupa, maka gambar pemandangan yang sama diambil berkali-kali. Rata-rata gambar kemudian digunakan untuk mengurangi derau.

Sumber: Dr. Sanjeev Kumar, *Mathematical Imaging Techniques*, Department of Mathematics, IIT Roorkee

A noisy image can be represented by

$$g(x, y) = f(x, y) + \eta(x, y),$$

where $\eta(x, y)$ denotes the noise in the image

Since the noise is random and the content $f(x, y)$ is fixed,

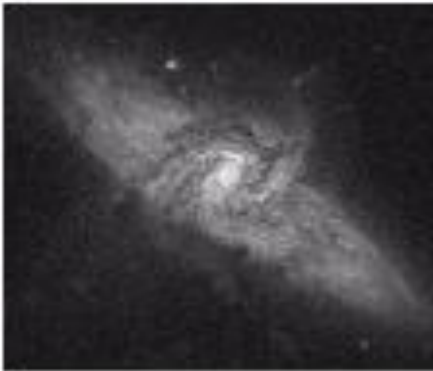
The noise can be removed by taking more noisy images of the same object and averaging them out

$$\bar{g}(x, y) = \frac{1}{K} \sum_{i=1}^K g_i(x, y),$$

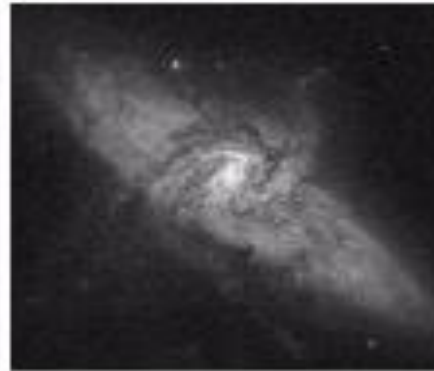
Original image



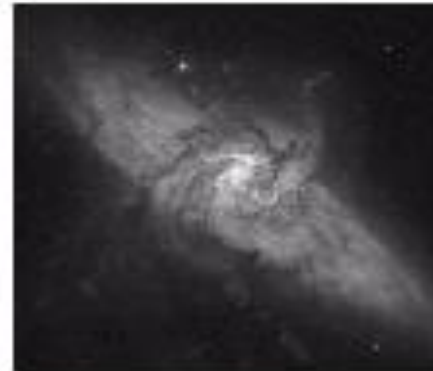
Noisy image



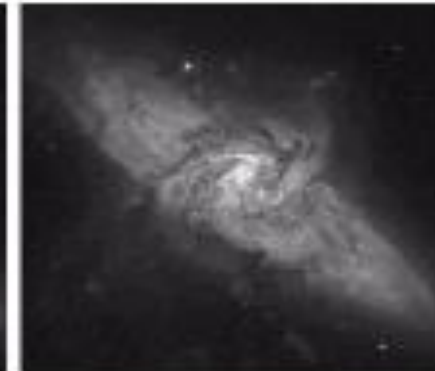
Result of averaging using 8 noise samples



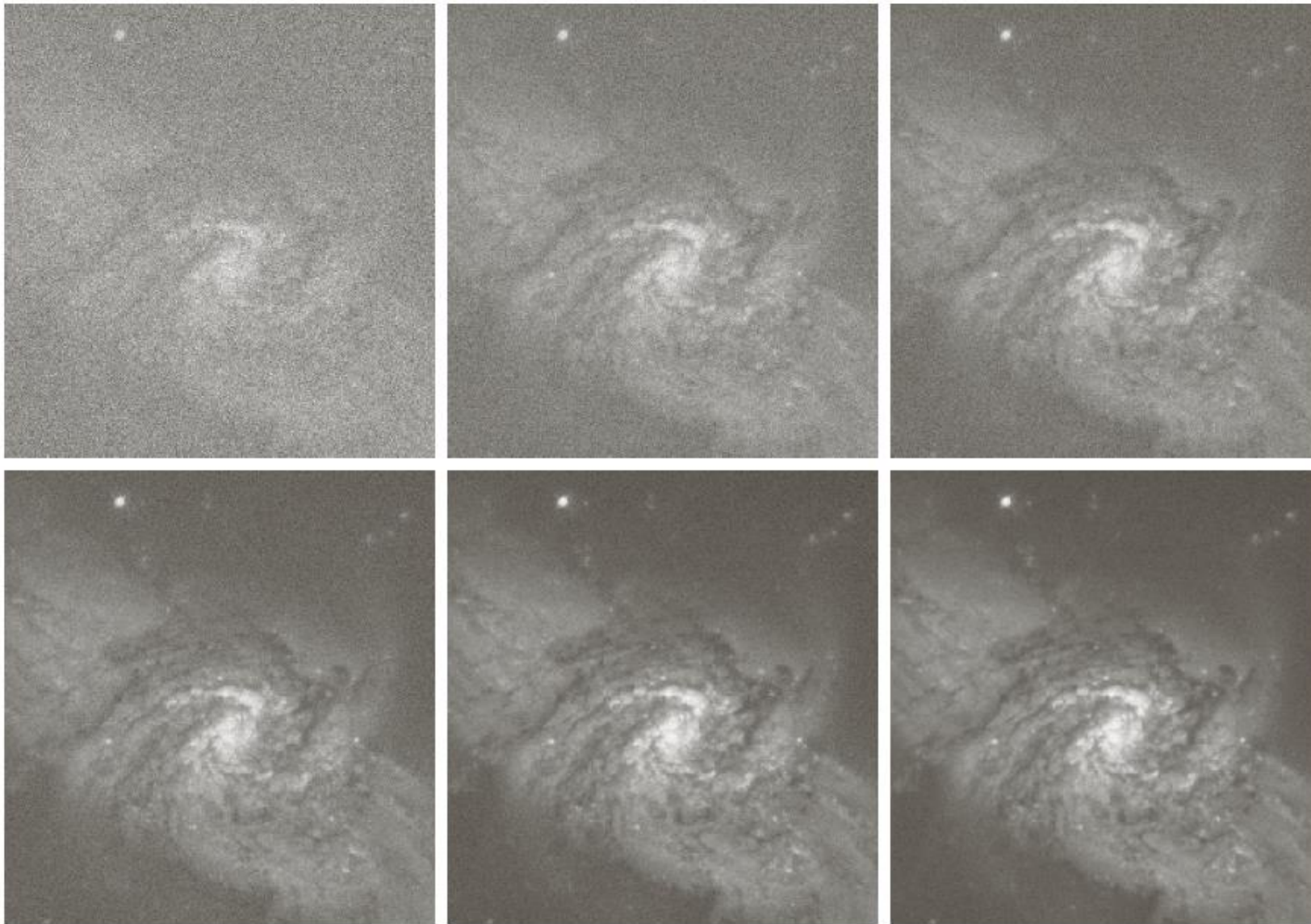
Using 16 noise samples



Using 64 noise samples



Using 128 noise samples



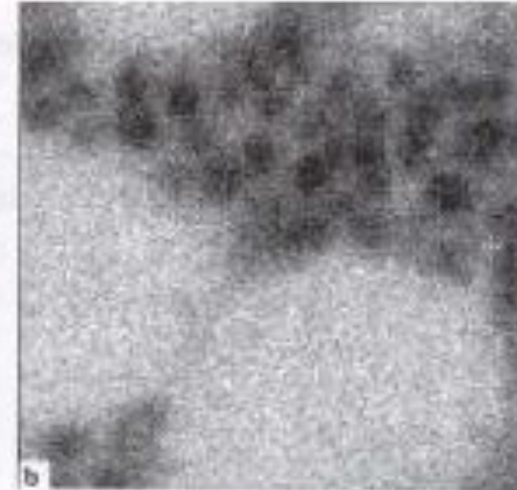
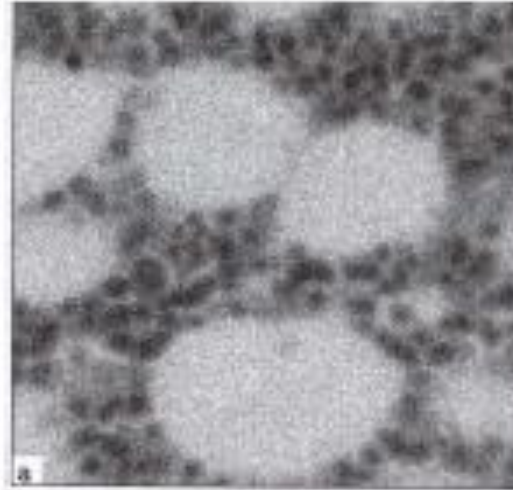
$$g(x, y) = f(x, y) + \eta(x, y)$$

$$\bar{g}(x, y) = \frac{1}{K} \sum_{i=1}^K g_i(x, y)$$

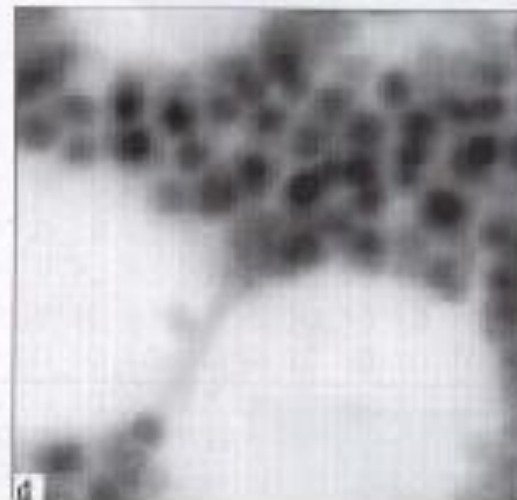
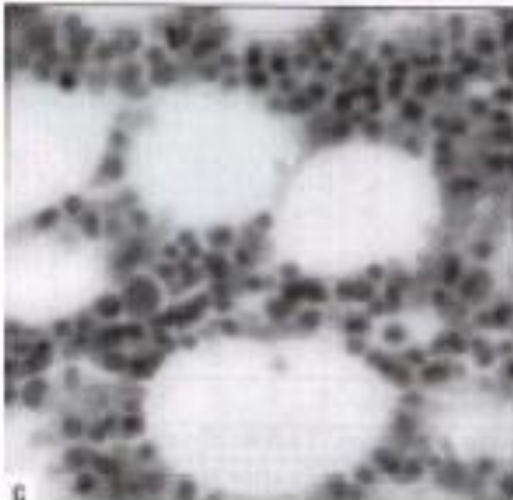
a b c
d e f

FIGURE 2.26 (a) Image of Galaxy Pair NGC 3314 corrupted by additive Gaussian noise. (b)–(f) Results of averaging 5, 10, 20, 50, and 100 noisy images, respectively. (Original image courtesy of NASA.)

Noisy image

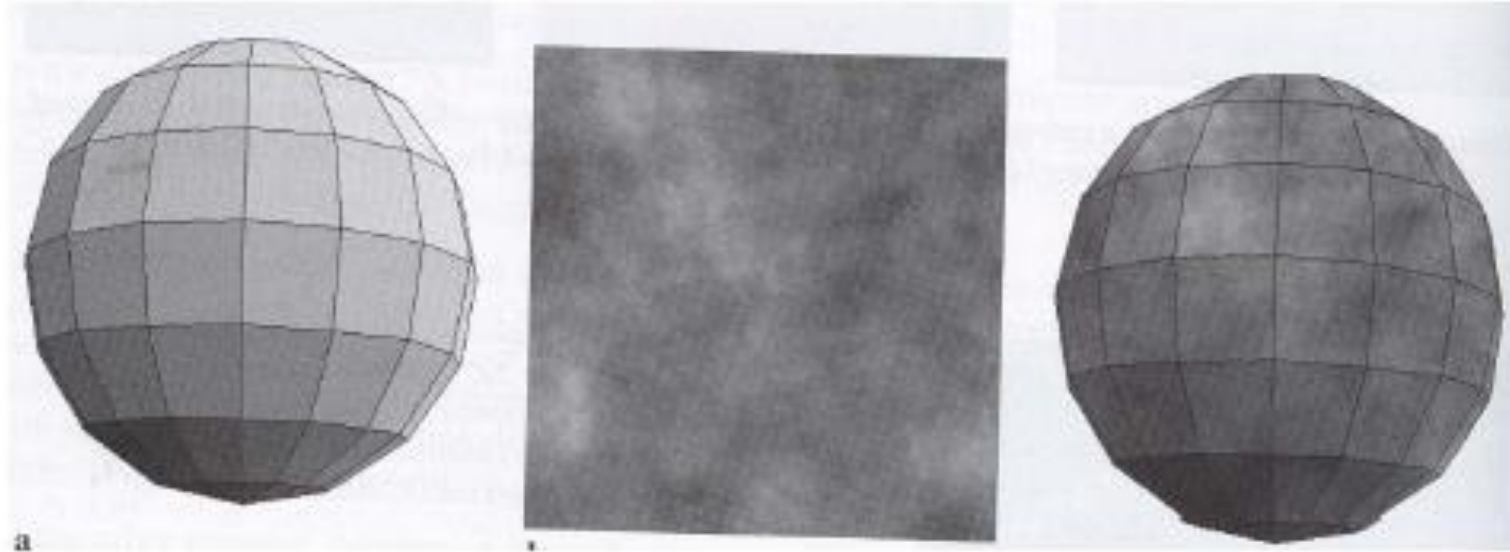


Noise
reduction by
averaging
256 samples



4. Operasi perkalian

Multiplication of images can be used for superimposing texture on an image



Smooth spherical surface
image

Texture to be
superimposed

output image



a b c

FIGURE 2.29 Shading correction. (a) Shaded SEM image of a tungsten filament and support, magnified approximately 130 times. (b) The shading pattern. (c) Product of (a) by the reciprocal of (b). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)