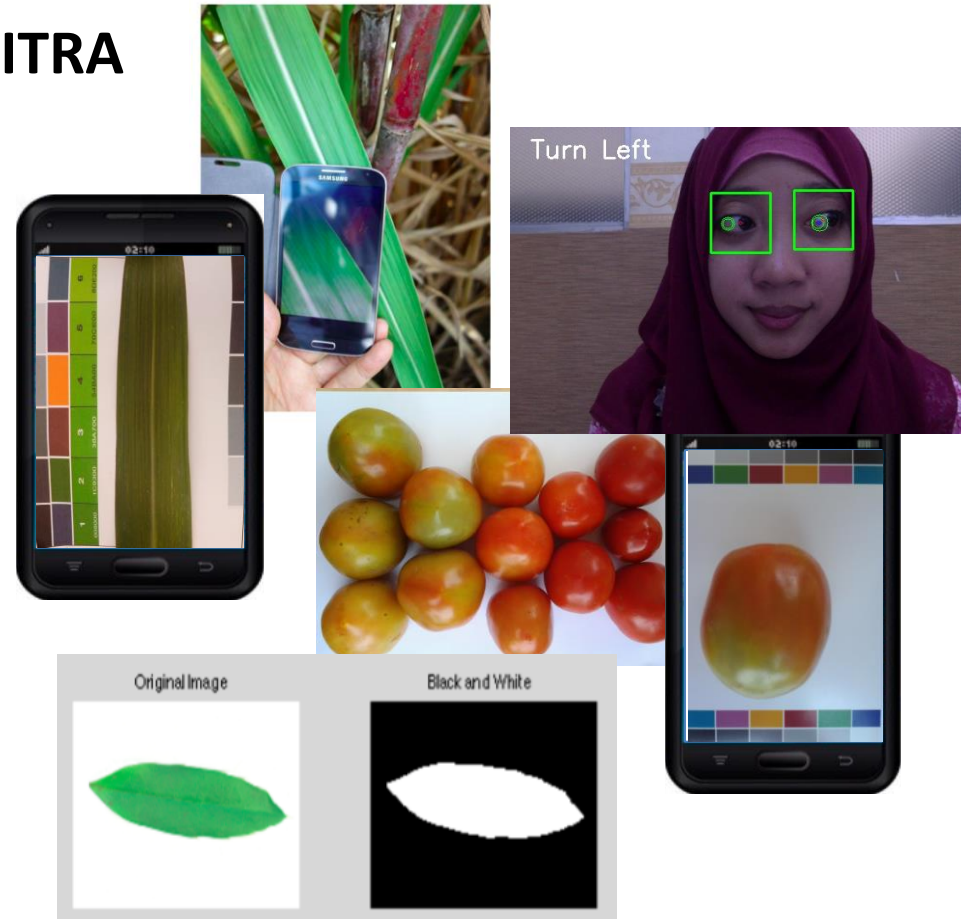


Pengolahan Citra Digital

3. PENINGKATAN KUALITAS CITRA IFK15036, 3 credits



Team Teaching :

1. Dr. Fitri Utaminingrum, S.T, M.T
2. Yuita Arum Sari, S.Kom, M.Kom

OUTLINE BAB-3

- ❑ Tentang Domain Spasial dan Domain Frekuensi Secara Umum. Yang kita gunakan pada pertemuan kali ini adalah pengolahan Citra pada domain spasial.
- ❑ *Intensity Transformation* (Transformasi Gray Level)
 - ❑ Transformasi pada Tingkat Keabuan Dasar
 - ❑ Transformasi Linear (Negative and Identity)
 - ❑ Transformasi Logaritmik (Log and Inverse Log)
 - ❑ Transformasi Power-Law (Pangkat)
 - ❑ Fungsi Transformasi Linear Sepotong-Sepotong
 - ❑ *Contrast stretching* dan *thresholding*
 - ❑ *Gray level slicing*
 - ❑ *Bit-plane slicing*
- ❑ Pemrosesan Histogram
 - ❑ Histogram Equalization (Perbaikan Global)
 - ❑ Perbaikan Lokal dengan Konsep Histogram Equalization
 - ❑ Penggunaan Nilai Statistik dari Histogram untuk Perbaikan Citra

- ✓ Transformasi citra, sesuai namanya, merupakan proses *perubahan bentuk* citra untuk mendapatkan suatu informasi tertentu.
- ✓ Sebuah citra digital di simbolkan dengan $f(x, y)$, dengan x dan y menyatakan baris dan kolom.
- ✓ Titik (x, y) menyatakan lokasi dari sebuah pixel didalam citra digital.
- ✓ Pixel-pixel disekeliling titik (x, y) disebut dengan pixel-pixel tetangga.
- ✓ Nilai pixel pada suatu titik (x, y) akan menjadi *intensity* atau disebut juga *gray-level* (level keabuan)
- ✓ Transformasi citra paling sederhana melibatkan 1 titik tetangga, tetapi lebih lanjut melibatkan banyak titik tetangga
- ✓ Transformasi dibagi menjadi 2 jenis yaitu :
 - **Transformasi pixel/transformasi geometris** → domain spasial
 - **Transformasi ruang/domain/space** → domain frekuensi
- ✓ Transformasi bisa digunakan untuk meningkatkan kualitas citra.

Transformasi Piksel/ Transformasi Geometris



- ❖ Transformasi piksel masih bermain di ruang/domain yang sama (**domain spasial**), hanya posisi piksel yang kadang diubah
- ❖ Contoh: rotasi, translasi, scaling, invers, shear, dll.
- ❖ Transformasi jenis ini relatif mudah diimplementasikan dan banyak aplikasi yang dapat melakukannya (Paint, ACDSee, dll)

- Transformasi ruang merupakan proses perubahan citra dari suatu ruang/domain ke ruang/domain lainnya.

Contoh : Ruang vektor. Salah satu basis yang merentang ruang vektor 2 dimensi adalah $[1\ 0]$ dan $[0\ 1]$. Artinya, semua vektor yang mungkin ada di ruang vektor 2 dimensi selalu dapat direpresentasikan sebagai kombinasi linier dari basis tersebut.

Domain Spasial dan Domain Frekuensi

- Teknik pemrosesan pada domain spasial didasarkan pada manipulasi piksel dalam citra secara langsung.
- Teknik pemrosesan pada domain frekuensi didasarkan pada manipulasi terhadap transformasi Fourier dari suatu citra.

Domain Spasial

Proses-proses pada domain spasial dinyatakan dengan ekspresi berikut:

$$g(x,y) = T [f(x,y)]$$

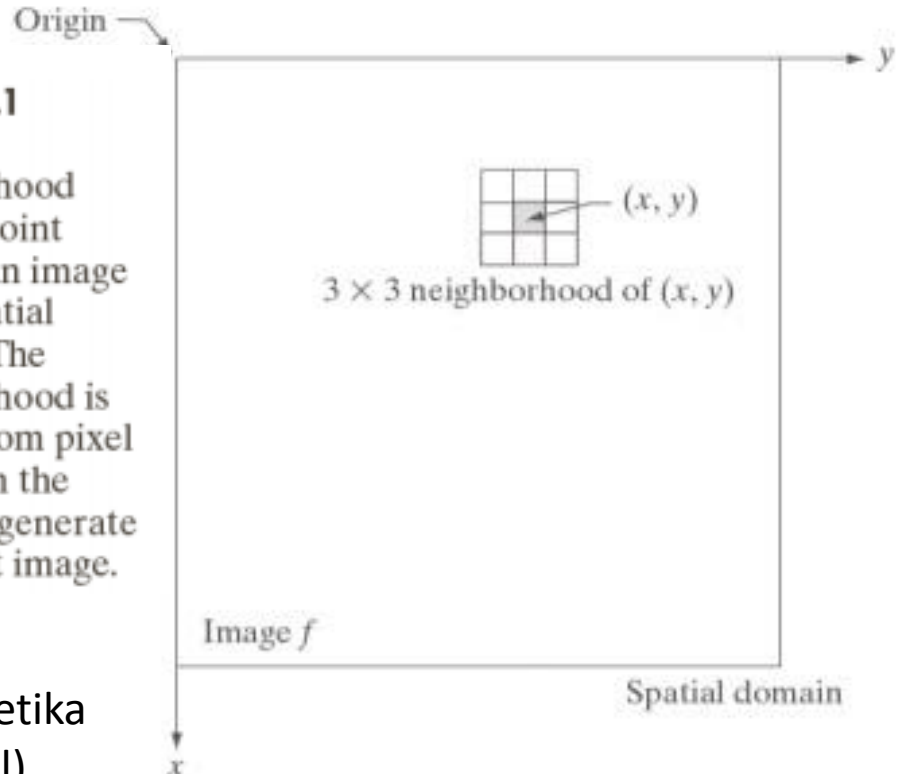
Dimana :

- $f(x,y)$ adalah citra input
- $g(x,y)$ adalah citra output
- T adalah operator terhadap f , yang didefinisikan pada ketetanggaan dari (x,y) .
- **T disebut juga sebagai *intensity transformation function***

Bentuk paling sederhana dari T adalah ketika ukuran neighborhood 1x1 (piksel tunggal).

FIGURE 3.1

A 3×3 neighborhood about a point (x, y) in an image in the spatial domain. The neighborhood is moved from pixel to pixel in the image to generate an output image.



Gambaran Transformasi Citra



$F(x,y)$

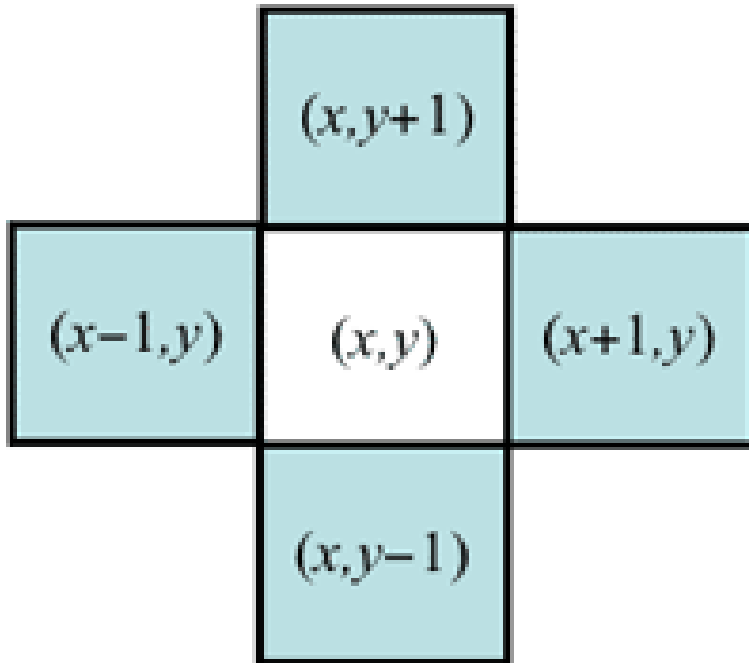


$G(x,y)$

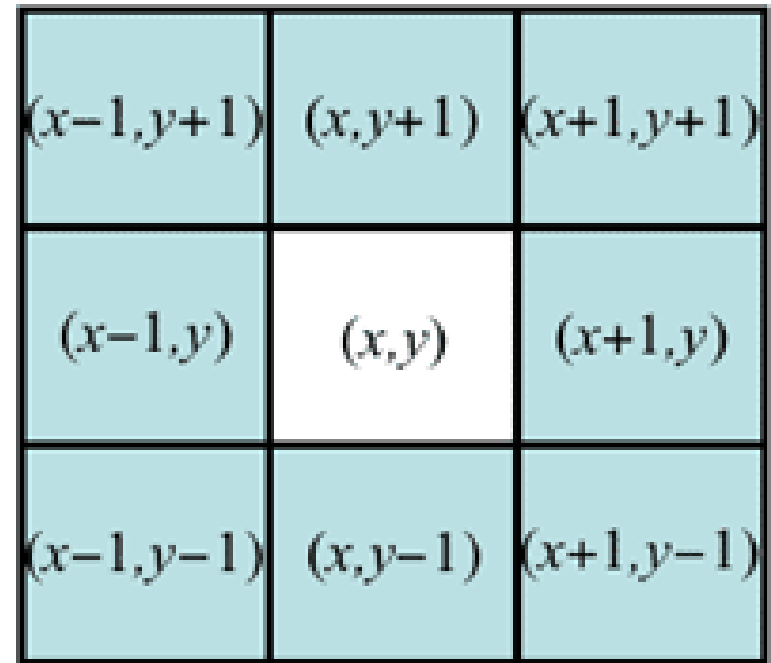


Transformasi citra paling sederhana melibatkan 1 titik tetangga, tetapi lebih lanjut melibatkan banyak titik tetangga

Operasi Ketetanggaan Piksel



4-neighbourhood



8-neighbourhood

Titik tetangga dari suatu titik (x,y) pada citra $F(x,y)$ adalah titik-titik terdekat dari posisi tersebut

Tranformasi Intensitas (*Intensity Transformation*)

- Transformasi gray level pada setiap piksel adalah sebuah citra input menjadi citra baru yang nampak lebih kontras.
- *Intensity transformation* merupakan teknik yang paling sederhana dari PCD.

Intensity Transformation Function **(Transformasi Intensitas)**

Transformasi pada Tingkat Keabuan Dasar

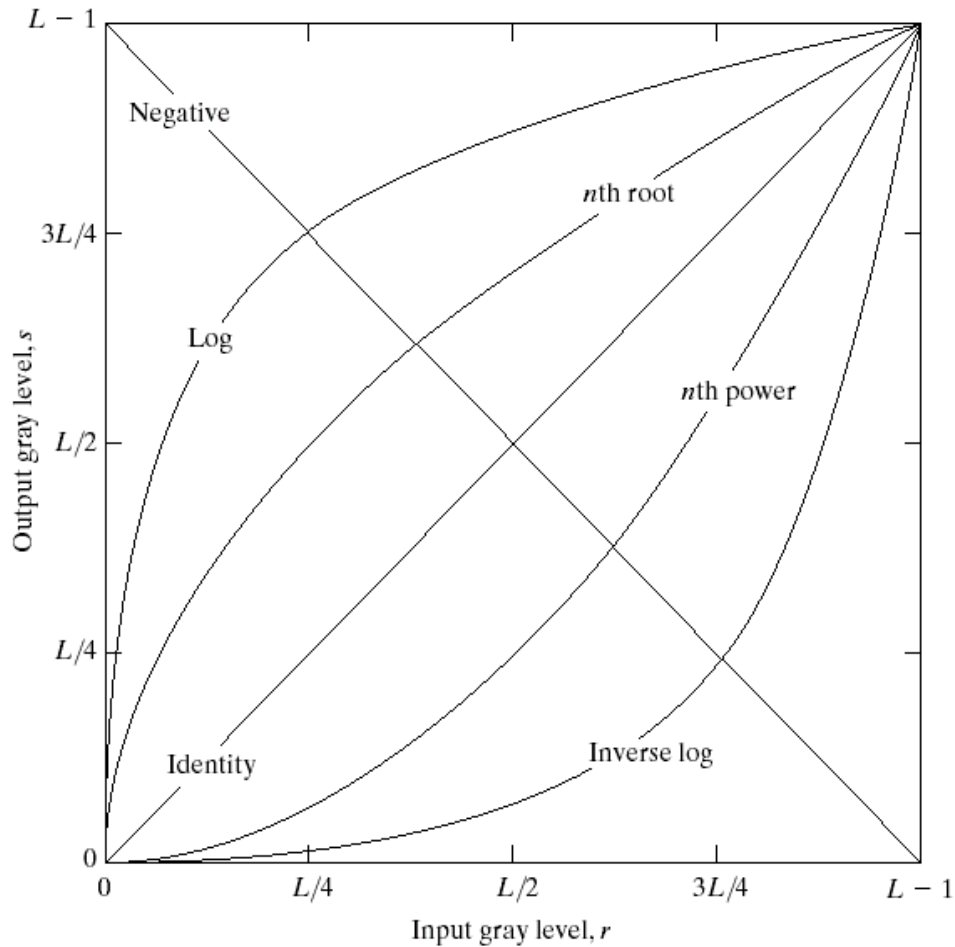
- ✓ Transformasi Linear (Negative and Identity)
- ✓ Transformasi Logaritmik (Log and Inverse Log)
- ✓ Transformasi Power-Law (Pangkat)

Fungsi Transformasi Linear Sepotong-Sepotong

- ✓ *Contrast strecting* dan *thresholding*
- ✓ *Gray level slicing*
- ✓ *Bit-plane slicing*

Jenis Transformasi Citra pada Tingkat Keabuan Dasar

FIGURE 3.3 Some basic gray-level transformation functions used for image enhancement.



**Transformasi Linear
(Negative dan Identity)**

**Transformasi Logaritmik
(Log dan Inverse Log)**

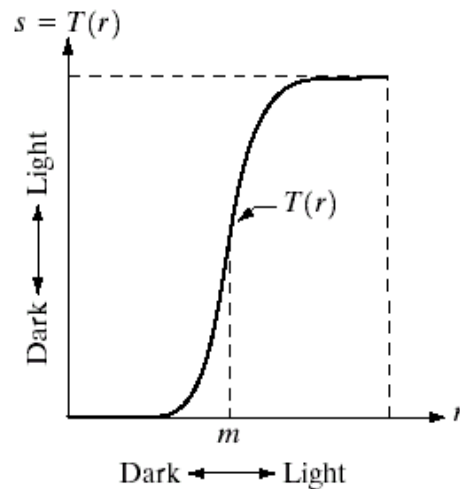
**Transformasi Power-Law
(n -th power dan n -th root power)**

Fungsi Transformasi Linear Sepotong- Sepotong (*Piecewise Linear Transformation Functions*)

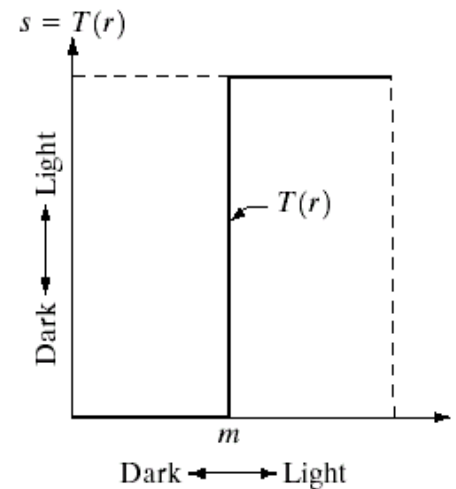
Contrast Streching dan Thresholding

Gray-level Slicing

Bit-plane slicing



(a) Contrast stretching function



(b) Thresholding function

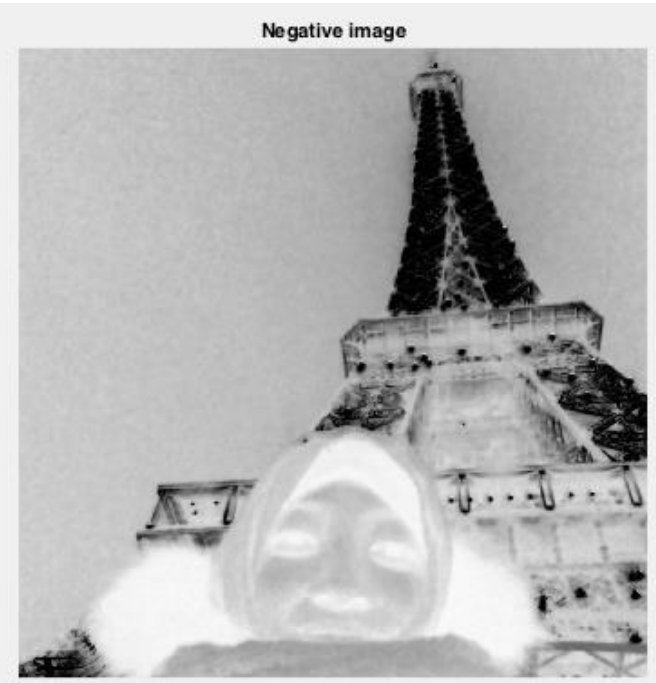
Transformasi Linear: *Image Negative and Identity*

- Secara visual *image negative* adalah kebalikan dari citra aslinya. Jika intensitas dari citra terang maka akan dibalik menjadi gelap dan begitu juga sebaliknya
- Sebuah *intensity-level* dalam *range* $[0, L-1]$, Misal untuk 8-bit memiliki nilai $L=256$, sehingga *memiliki intensity-level* dalam *ring* 0 sampai dengan 255 atau 0 s/d $L-1$.
- Untuk mendapatkan nilai negatifnya dapat dilakukan dengan mengurangi nilai *intensity level* maksimalnya $(L-1)$ dengan nilai pixel masukan (r). Atau dapat dirumuskan dengan:

$$s = (L - 1) - r$$

Transformasi Linear: Image Negative

OUTPUT



```
clear all
clc
close all
r = imread('grey_parisnight.jpg');
L = 255
negiImg= L-1-r; %s=L-1-r
imshow(negiImg);
subplot(1,2,1);imshow(r);title('Original image');
subplot(1,2,2);imshow(negiImg);title('Negative image');
```

Hasil Citra Negatif

Original image



Negative image



Original image



Negative image



Hasil citra negatif biasanya digunakan untuk :

- Citra pada rekam medis seperti rontgen.
- Meningkatkan warna putih atau abu-abu pada area citra yang gelap.
- Membuat efek foto klise pada video editing,dll.

Transformasi Logaritmik

- Transformasi logaritmik didefinisikan dengan:

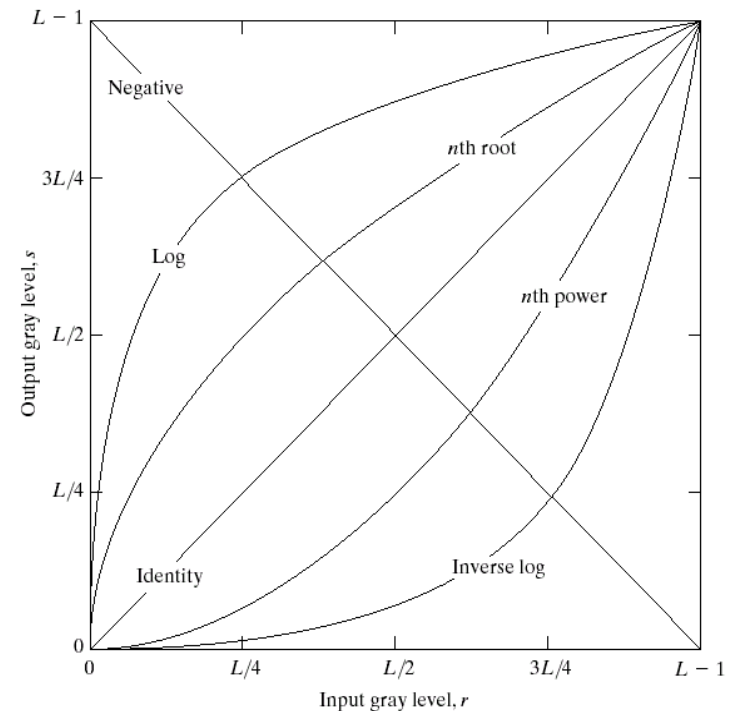
$$G = c \text{ Log } (F+1)$$

- Transformasi Inverse Logaritmik didefinisikan dengan:

$$G = c \text{ Log } (L-F+1)$$

- ✓ G adalah citra hasil dan F adalah citra asal
- ✓ c adalah konstanta yang dipasang sebagai efek perubahan kontras

- Transformasi log memetakan rentang yang sempit dari nilai-nilai tingkat keabuan gelap pada citra input ke dalam rentang yang lebih luas pada citra output. Kebalikannya berlaku untuk tingkat keabuan terang.
- Transformasi log memperbanyak jumlah piksel bernilai gelap dan mengurangi jumlah piksel bernilai terang. Pada transformasi inverse log, yang terjadi adalah kebalikannya.



Hasil Transformasi Logaritmik

Transformasi logaritmik $G = c \text{Log} (F+1)$, dengan $c = 0.5$ dihasilkan gambar sebagai berikut :

Original image



Logaritmik image



Transformasi Pangkat

Transformasi pangkat dirumuskan :

$$s = cr^\gamma$$

dengan c dan γ adalah konstanta positif.

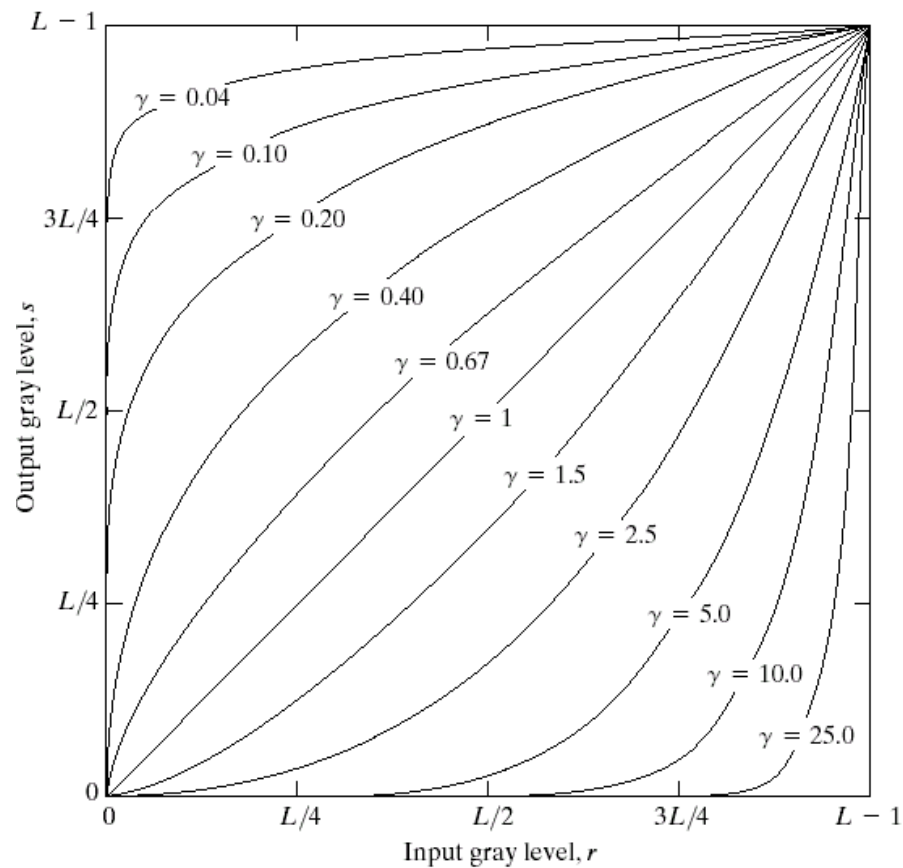


FIGURE 3.6 Plots of the equation $s = cr^\gamma$ for various values of γ ($c = 1$ in all cases).

Transformasi Pangkat

Power Image with gama 2.5



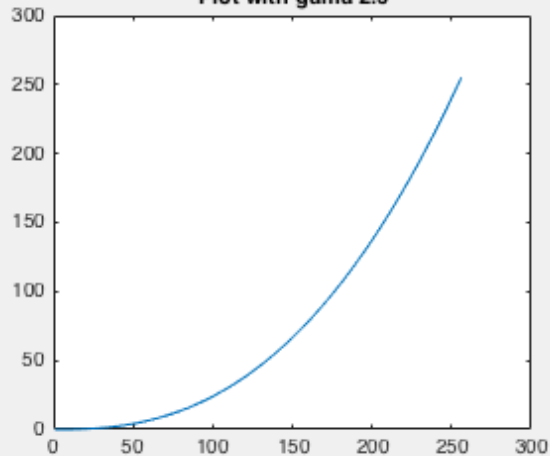
Power Image with gama = 1



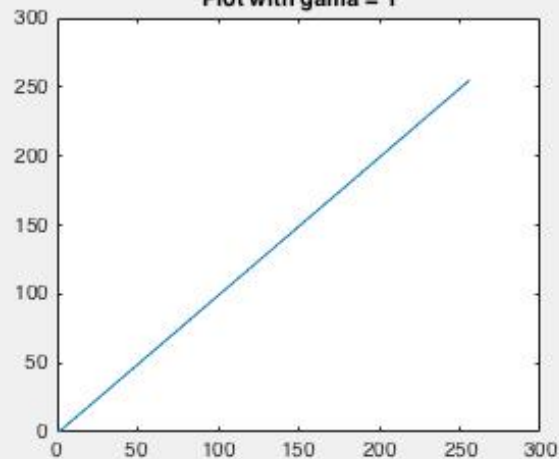
Power Image with gama = 0



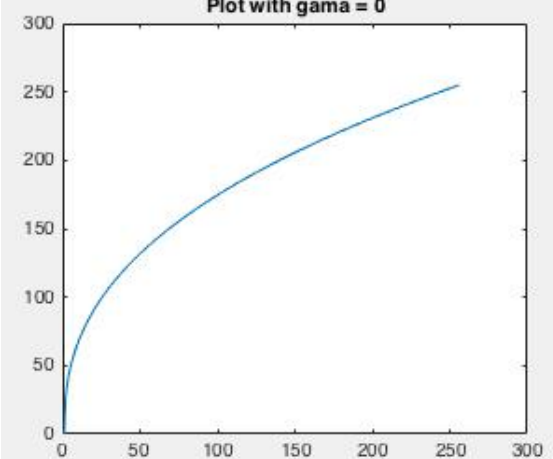
Plot with gama 2.5



Plot with gama = 1



Plot with gama = 0



Fungsi Transformasi Linear Sepotong-Sepotong

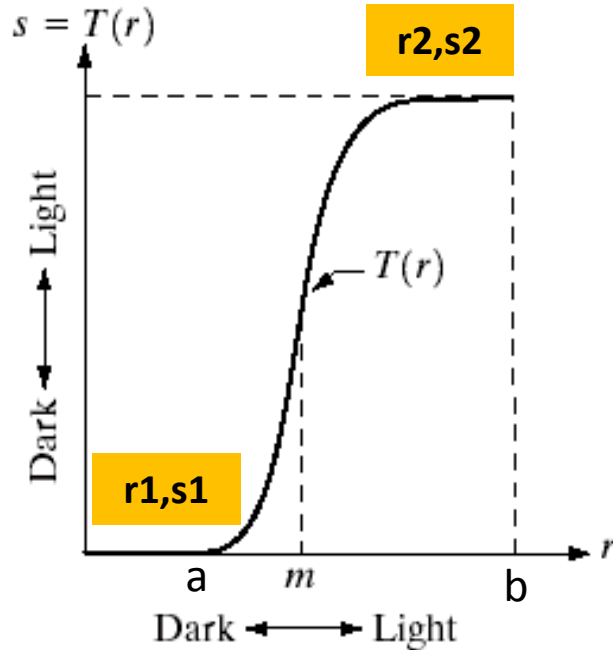
- Selain tiga fungsi transformasi dasar yang dibahas sebelumnya, fungsi transformasi linear sepotong-sepotong juga biasa digunakan.
- Keuntungannya, bentuk dari fungsi sepotong-sepotong bisa lebih kompleks dibandingkan fungsi transformasi dasar.

Contrast Streching dan Thresholding

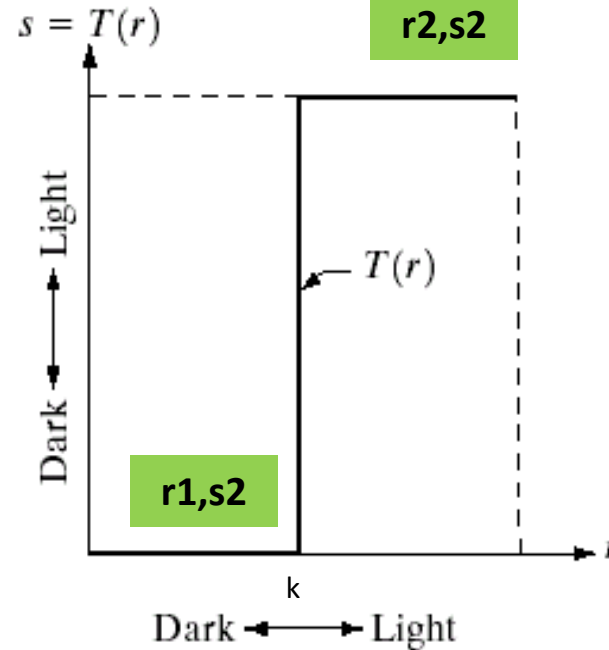
Gray-level Slicing

Bit-plane slicing

Contrast Streching dan Thresholding



(a) Contrast stretching function



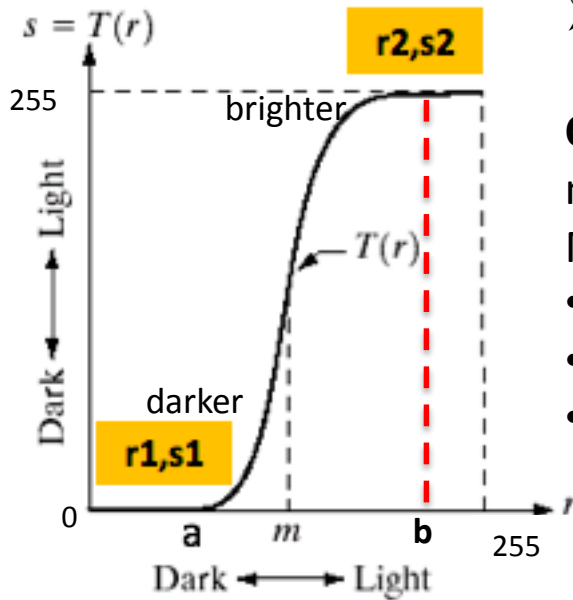
(b) Thresholding function

Misalnya $a=r_{\min}$, $b=r_{\max}$, dan $k=\text{intensity}$

Contrast Stretching $(r_1, s_1) = (r_{\min}, 0)$; $(r_2, s_2) = (r_{\max}, L-1)$

Thresholding $(r_1, s_1) = (k, 0)$; $(r_2, s_2) = (k, L-1)$

Contrast Stretching



➤ *Contrast Stretching* biasanya digunakan untuk menajamkan gambar

Contoh : Pada grafik diatas diketahui misalnya $a=90$, $b =180$, $m=100$

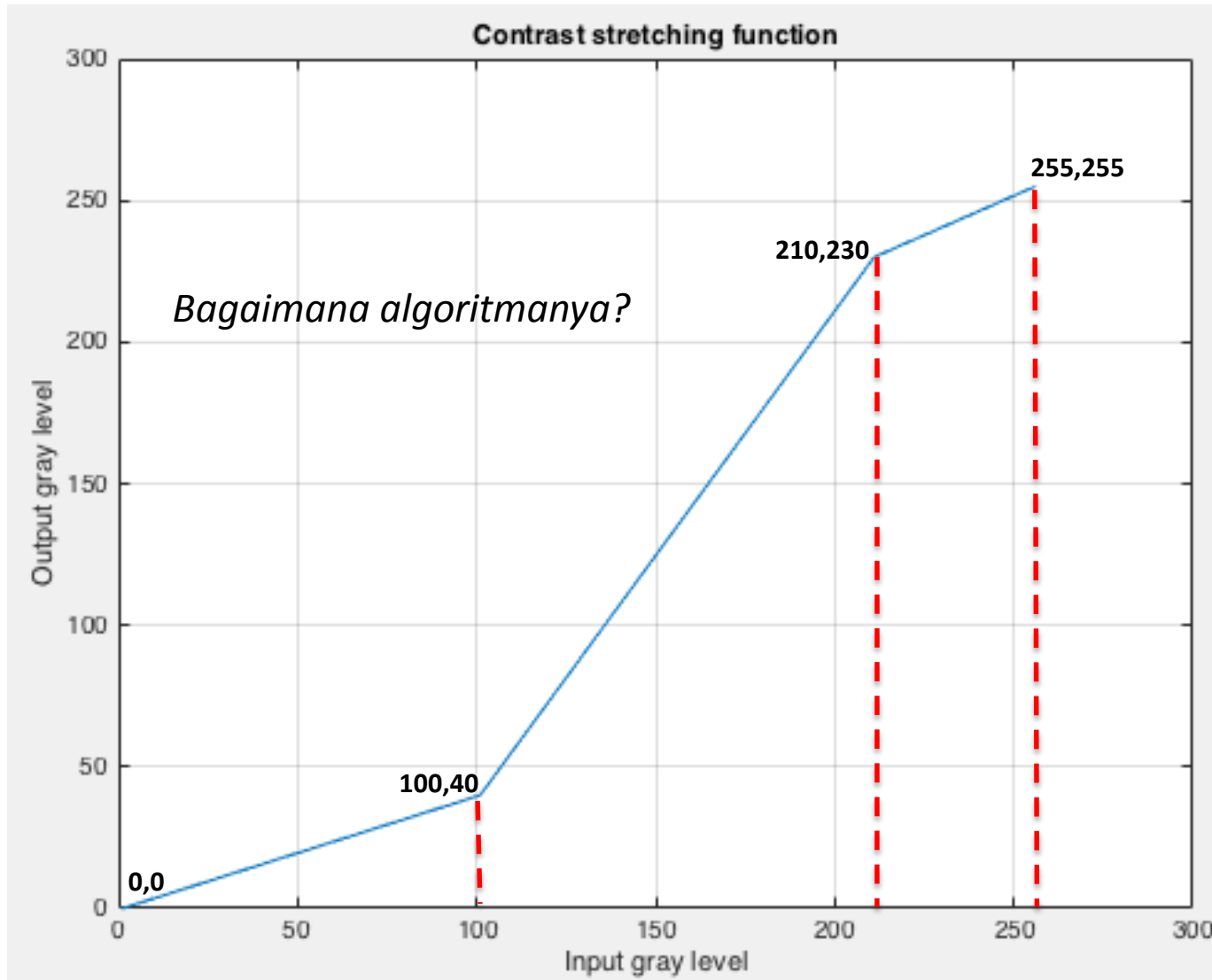
Maka :

- Jika r diatas 180, maka nilai s akan semakin mendekati 255
- Jika r dibawah 90 maka nilai s akan menjadi 0
- Jika r diantara 90 dan 180 maka representasi dari T adalah
 - Jika $r < 100$, maka s menjadi semakin mendekati 0 (gelap)
 - Jika $r > 100$, maka s menjadi semakin mendekati 255 (terang)

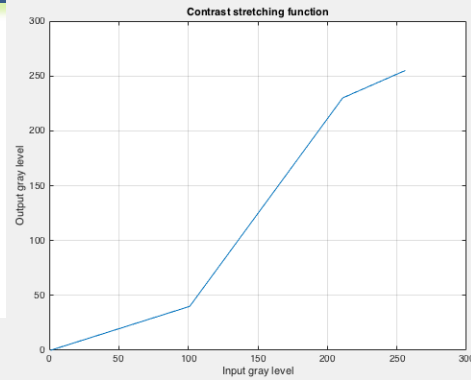
- Citra dengan kontras rendah bisa disebabkan oleh kurangnya pencahayaan, kurangnya rentang dinamis dari peralatan sensor citra, atau setting lensa yang salah pada saat pengambilan citra.
- Ide dibalik “contrast stretching” adalah meningkatkan rentang dinamis tingkat keabuan dari citra.

Contrast stretching → Pixel yang terang akan semakin terang dan pixel yang gelap akan semakin gelap : **higher contrast image**

Fungsi Contrast Streching



Hasil Fungsi Contrast Streching



Original image



Contrast stretching



Thresholding

- Thresholding adalah sebuah **proses mengubah citra keabuan menjadi citra biner**
- Thresholding merupakan teknik sederhana yang biasa diterapkan untuk **segmentasi citra**, sebagai bentuk dari teknik *preprocessing* citra

Contoh citra biner hasil *thresholding*

Original Image



Grey Image



Thresholding Image



Original Image



Grey Image



Thresholding Image



Contoh Proses Thresholding

```
abu2 = warna2abu('text.jpg');  
[a,b] = size(abu2); threshold = 175 ; % nilai ini bisa berubah  
binary = zeros(a,b); % inisialisasi  
for baris=1:a  
    for kolom=1:b  
        if abu2(baris,kolom) >= threshold  
            binary(baris,kolom) = 255;  
        else  
            binary(baris,kolom) = 0;  
        end  
    end  
end  
End
```

Dalam contoh ini *thresholding* bisa digunakan untuk mengubah gambar text berwarna ke dalam gambar text hitam-putih. Text hitam putih tersebut bisa digunakan untuk pengolahan bahasa alami, *text mining*, dsb

1.3.3 Imaging in the Ultraviolet Band

Applications of ultraviolet "light" are varied. They include lithography inspection, microscopy, lasers, biological imaging, and astronomical imaging. We illustrate imaging in this band with examples from microscopy and astronomy.

Ultraviolet light is used in fluorescence microscopy, one of the interesting areas of microscopy. Fluorescence is a phenomenon discovered in the middle of the nineteenth century, when it was first observed that fluor spar fluoresces when ultraviolet light is directed upon it. The light itself is not visible, but when a photon of ultraviolet radiation strikes an electron in an atom of a fluorescent material, it elevates the electron to a higher energy level. Subsequently, the excited electron relaxes to a lower energy level and emits light in the form of a lower-energy photon in the visible (red) light spectrum. The basic task of the fluorescence microscope is to use an excitation light



1.3.3 Imaging in the Ultraviolet Band

Applications of ultraviolet "light" are varied. They include lithography inspection, microscopy, lasers, biological imaging, and astronomical imaging. We illustrate imaging in this band with examples from microscopy and astronomy.

Ultraviolet light is used in fluorescence microscopy, one of the interesting areas of microscopy. Fluorescence is a phenomenon discovered in the middle of the nineteenth century, when it was first observed that fluor spar fluoresces when ultraviolet light is directed upon it. The light itself is not visible, but when a photon of ultraviolet radiation strikes an electron in an atom of a fluorescent material, it elevates the electron to a higher energy level. Subsequently, the excited electron relaxes to a lower energy level and emits light in the form of a lower-energy photon in the visible (red) light spectrum. The basic task of the fluorescence microscope is to use an excitation light

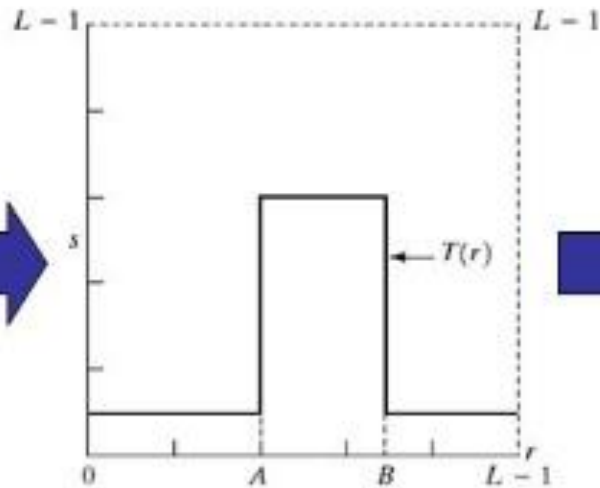
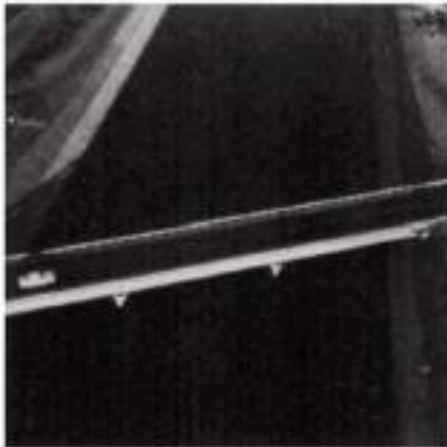
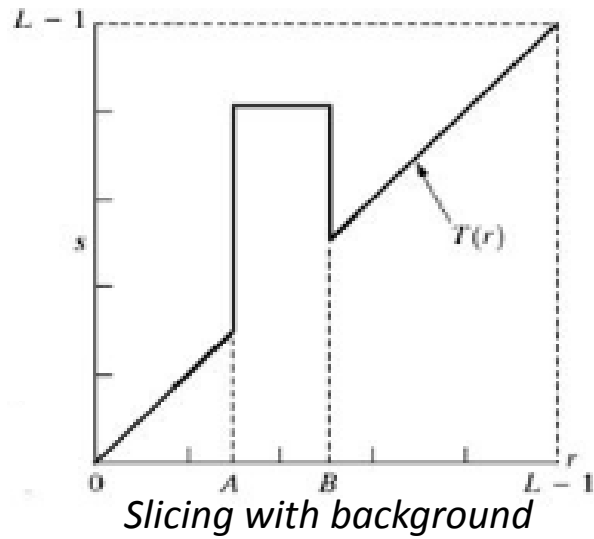
Gray Level Slicing

Terkadang diperlukan untuk menonjolkan rentang tertentu dari tingkat keabuan yang ada dalam citra. Misalnya, menonjolkan gumpalan air yang ada pada citra satelit dan menonjolkan cacat yang ada pada citra sinar X.

Salah satu cara yang bisa dilakukan adalah dengan menampilkan secara lebih terang semua tingkat keabuan dalam range yang ingin ditonjolkan, dan menampilkan secara lebih gelap semua tingkat keabuan lainnya.

Cara lain adalah dengan menampilkan secara lebih terang semua tingkat keabuan dalam range yang ingin ditonjolkan, dengan tetap mempertahankan proporsi tingkat keabuan lainnya.

Gray Level Slicing



Contoh Hasil *Gray Level Slicing*

```
abu2=imread(warna2abu('finlandia.jpg'));  
z=abu2;  
[m,n]=size(abu2);  
mask = (z > 5) & (z < 120);  
z(mask) = 255;  
Imshow(z, [])
```

Original image



grey level slicing with background

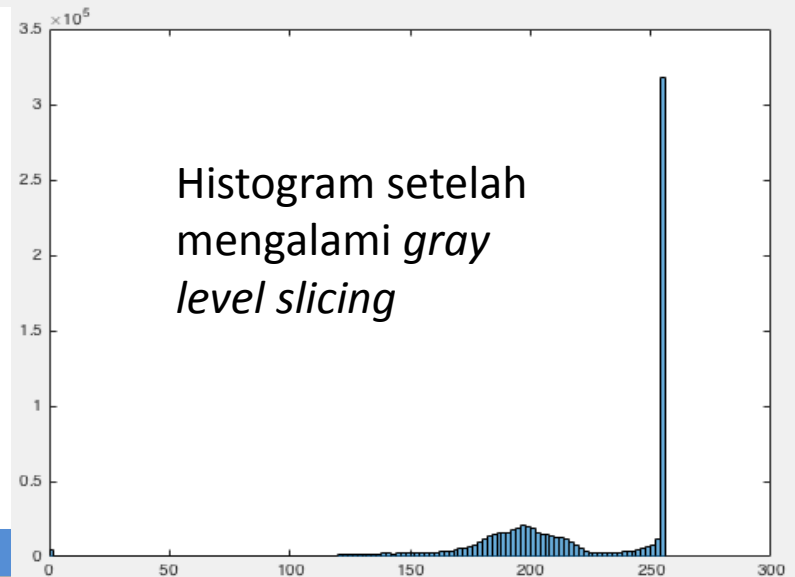
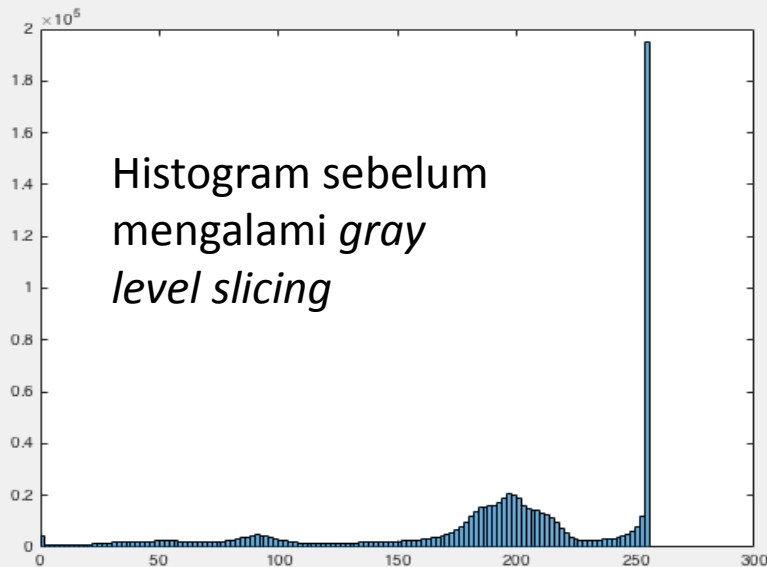


Contoh Hasil *Gray Level Slicing*

Original image



grey level slicing with background



- Selain menonjolkan range tingkat keabuan tertentu, terkadang diperhatikan juga bagaimana menonjolkan kontribusi dari bit tertentu pada kemunculan citra.
- Misalkan intensitas tiap piksel dalam citra dinyatakan dengan 8 bit. Sehingga citra tersusun atas 8 bidang 1-bit, mulai dari bidang bit 0 untuk “least significant bit” (LSB) sampai bidang bit 7 untuk “the most significant bit” (MSB).

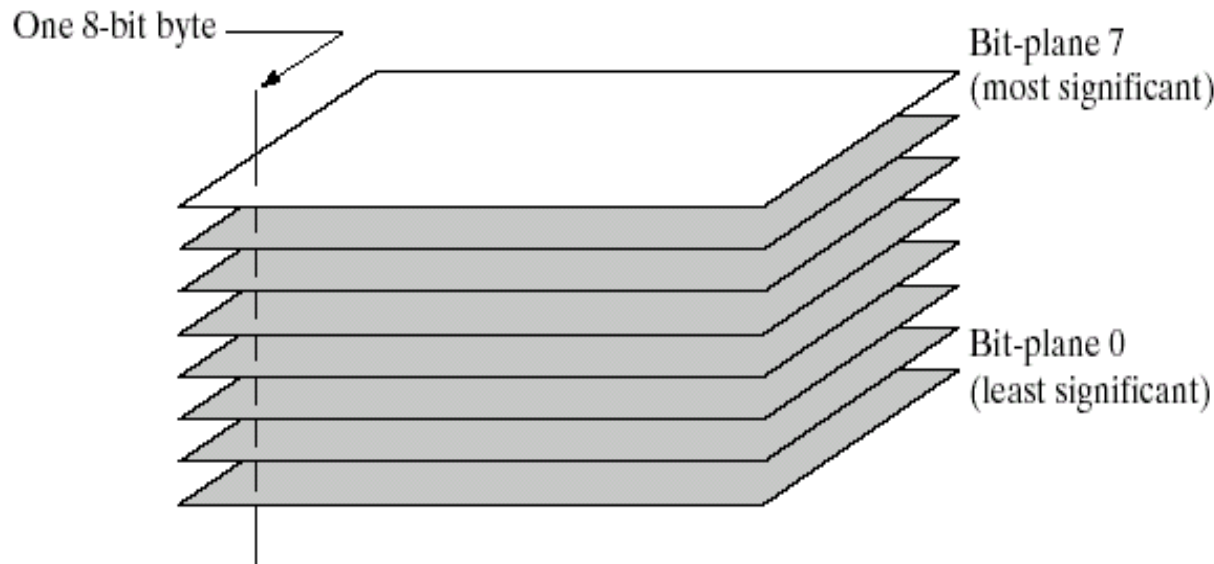


FIGURE 3.12
Bit-plane
representation of
an 8-bit image.

- Biasa digunakan untuk teknik kompresi pada gambar.

Proses Pada *Bit-Plane Slicing*



1. Asumsikan piksel dengan representasi nilai matriks A seperti berikut :

$$A = \begin{array}{|c|c|c|} \hline 229 & 230 & 231 \\ \hline 224 & 225 & 226 \\ \hline 198 & 199 & 200 \\ \hline \end{array}$$

2. Ubah nilai intensitas matrix A ke dalam bentuk biner. Misalnya untuk 229 diubah ke dalam biner menjadi 11100101. Jadi dalam 1 piksel gambar terdiri dari 8 bit biner.

$$A_{\text{biner}} = \begin{array}{|c|c|c|} \hline 11100101 & 11100110 & 11100111 \\ \hline 11100000 & 11100001 & 11100010 \\ \hline 11000110 & 11000111 & 11001000 \\ \hline \end{array}$$

Bit-plane 1

Untuk Citra 1-bit plane mengandung bit terendah (LSB) dari keseluruhan nilai yang ada di dalam piksel

| | | |
|----------|----------|----------|
| 11100101 | 11100110 | 11100111 |
| 11100000 | 11100001 | 11100010 |
| 11000110 | 11000111 | 11001000 |

| | | |
|-----|-----|-----|
| 229 | 230 | 231 |
| 224 | 225 | 226 |
| 198 | 199 | 200 |

Citra Awal



| | | |
|---|---|---|
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 0 | 1 | 0 |

Citra Akhir

Proses Pada *Bit-Plane 8 Slicing*

Bit-plane 8

Untuk Citra 8-bit plane mengandung bit tertinggi (MSB) dari keseluruhan nilai yang ada di dalam piksel

| | | |
|----------|----------|----------|
| 11100101 | 11100110 | 11100111 |
| 11100000 | 11100001 | 11100010 |
| 11000110 | 11000111 | 11001000 |

| | | |
|-----|-----|-----|
| 229 | 230 | 231 |
| 224 | 225 | 226 |
| 198 | 199 | 200 |

Citra Awal



| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

Citra Akhir

Contoh Hasil dari *Bit-Plane Slicing*

Original 8 bit image



Bitplane 1



Bitplane 2



Bitplane 3



Bitplane 4



Bitplane 5



Bitplane 6



Bitplane 7

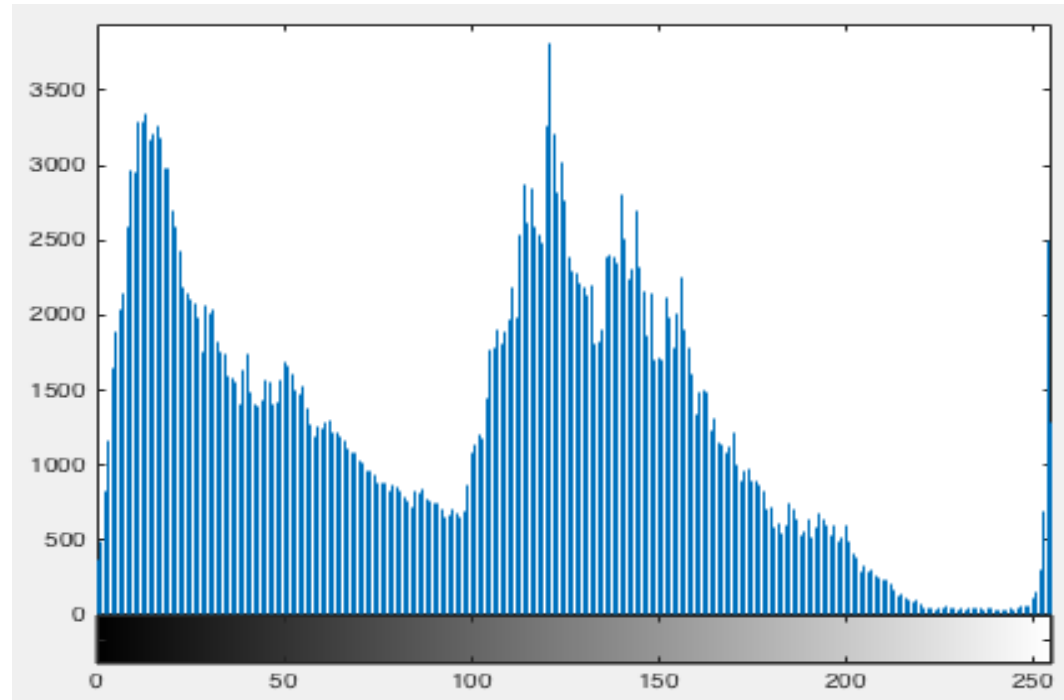


Bitplane 8



Pemrosesan Histogram

- Kita sudah membahas definisi secara umum dari Histogram pada pertemuan ke-2
- Histogram adalah dasar dari sejumlah teknik pemrosesan citra pada domain spasial, seperti perbaikan, kompresi dan segmentasi citra.



- Histogram digunakan untuk melihat frekuensi kemunculan suatu nilai intensitas pixel melalui grafik (bar).

Normalisasi Histogram

➤ **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 $p(r_k) = n_k/MN$, 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**

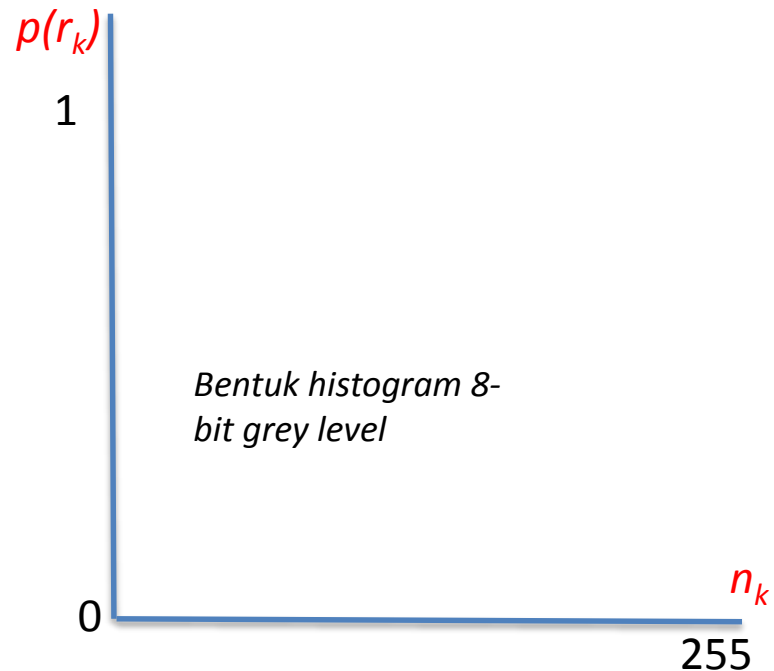
➤ r_k adalah tingkat keabuan ke- k

➤ n_k adalah jumlah piksel dalam citra yang memiliki tingkat keabuan r_k

➤ 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).

Rumus normalisasi histogram :

$$p(r_k) = \frac{n_k}{MN}$$



Proses Perhitungan Manual Histogram

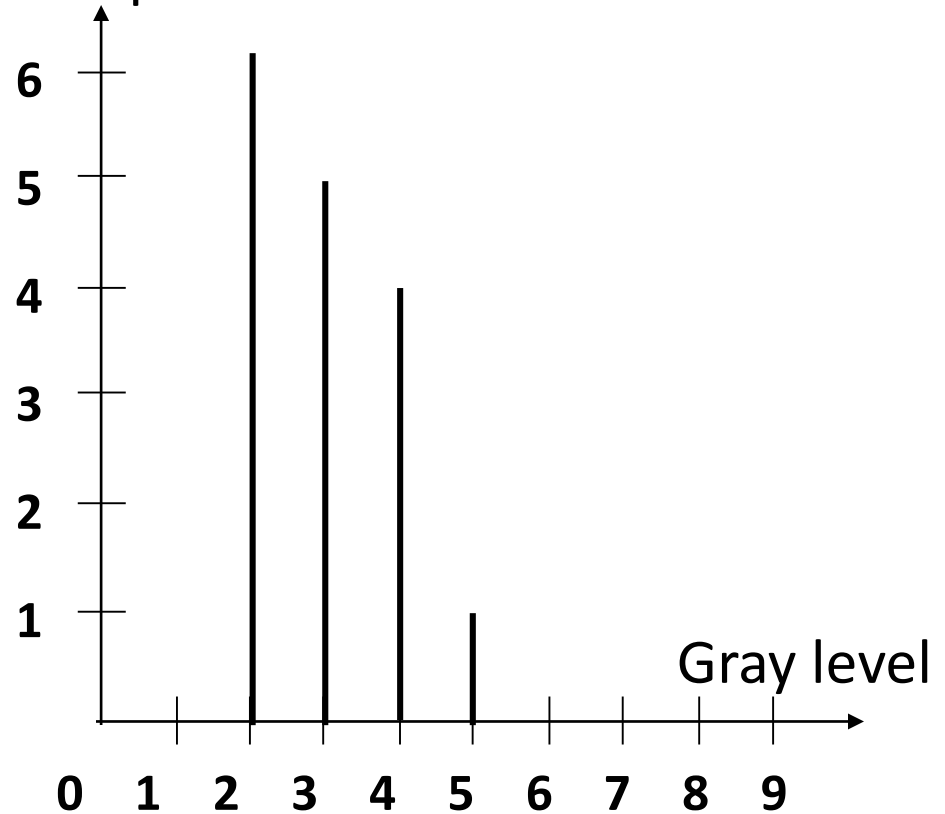
Contoh :

| | | | |
|---|---|---|---|
| 2 | 3 | 3 | 2 |
| 4 | 2 | 4 | 3 |
| 3 | 2 | 3 | 5 |
| 2 | 4 | 2 | 4 |

4x4 image

Gray scale = [0,9]

No. of pixels



histogram

- ❖ “Histogram equalization” digunakan untuk memperlebar range tingkat keabuan, sehingga akan meningkatkan kekontrasan citra.
- ❖ Histogram Equalization disebut juga dengan perbaikan global

Transformasi citra menggunakan histogram *equalization* dijelaskan sebagai berikut:

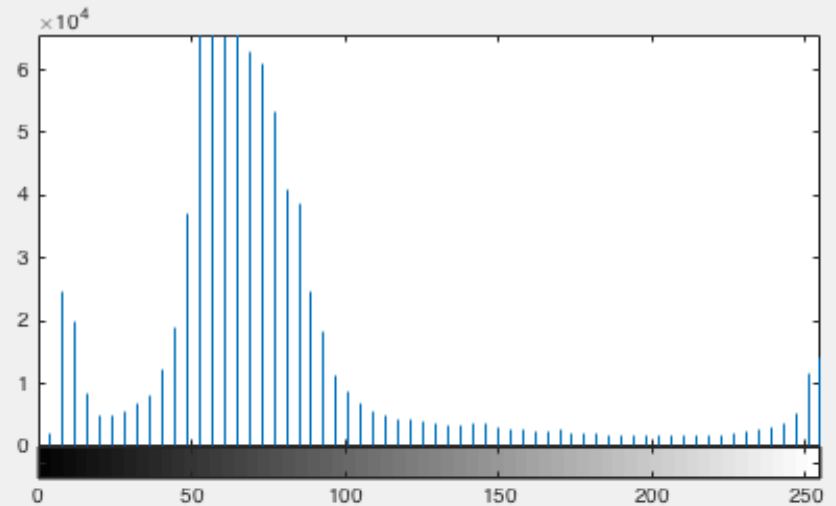
$$\begin{aligned} s_k &= T(r_k) = \sum_{j=0}^k p_r(r_j) \\ &= \sum_{j=0}^k \frac{n_j}{n} \end{aligned}$$

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

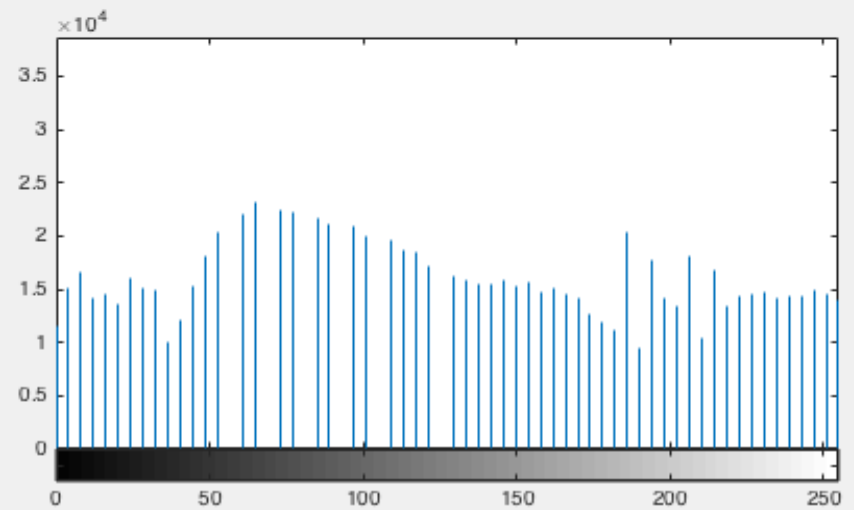
disebut “histogram equalization” atau “histogram linearization”.

Histogram Equalization

original image



histogram equalization



Proses Perhitungan Manual Histogram Equalization

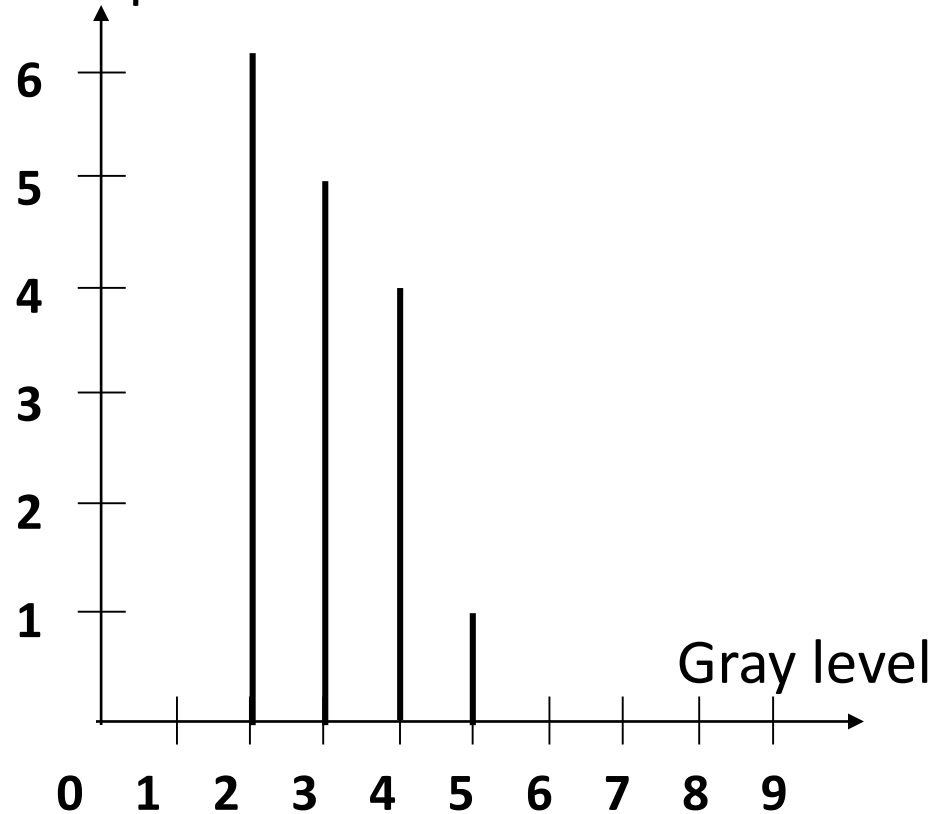
Contoh :

| | | | |
|---|---|---|---|
| 2 | 3 | 3 | 2 |
| 4 | 2 | 4 | 3 |
| 3 | 2 | 3 | 5 |
| 2 | 4 | 2 | 4 |

4x4 image

Gray scale = [0,9]

No. of pixels



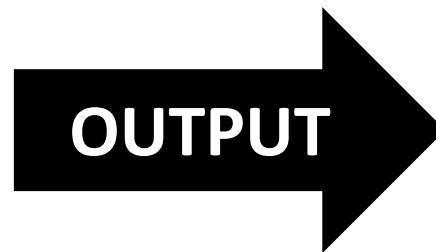
histogram

Proses Perhitungan Manual Histogram Equalization

| | | | | | | | | | | |
|----------------------------------|---|---|-----------|-----------|-----------|-------|-------|-------|-------|-------|
| 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 |
| $\sum_{j=0}^k n_j$ | 0 | 0 | 6 | 11 | 15 | 16 | 16 | 16 | 16 | 16 |
| $s = \sum_{j=0}^k \frac{n_j}{n}$ | 0 | 0 | 6 / 16 | 11/16 | 15/16 | 16/16 | 16/16 | 16/16 | 16/16 | 16/16 |
| s x 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 |

Citra awal



| | | | |
|---|---|---|---|
| 3 | 6 | 6 | 3 |
| 8 | 3 | 8 | 6 |
| 6 | 3 | 6 | 9 |
| 3 | 8 | 3 | 8 |

Citra hasil histogram equalization

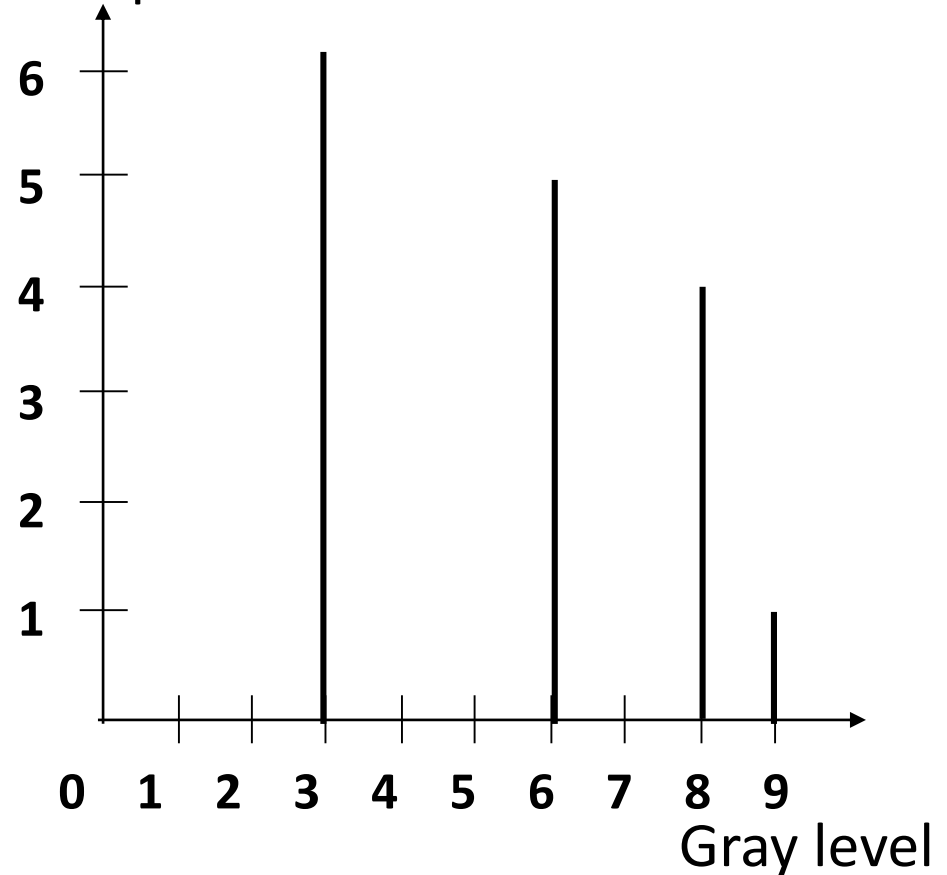
Hasil Histogram dari Citra yang Telah Dilakukan *Histogram Equalization*

| | | | |
|---|---|---|---|
| 3 | 6 | 6 | 3 |
| 8 | 3 | 8 | 6 |
| 6 | 3 | 6 | 9 |
| 3 | 8 | 3 | 8 |

Output image

Gray scale = [0,9]

No. of pixels



Latihan : Histogram Equalization

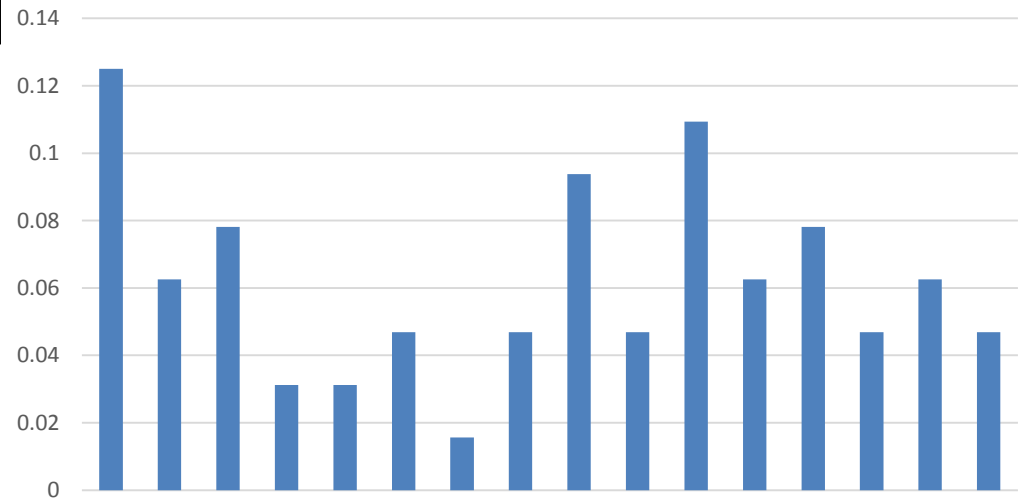
| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 2 | 1 | 0 | 10 | 11 | 14 | 13 | 12 |
| 15 | 13 | 11 | 10 | 9 | 9 | 8 | 7 |
| 4 | 5 | 0 | 0 | 1 | 0 | 2 | 2 |
| 0 | 2 | 3 | 4 | 5 | 13 | 10 | 14 |
| 12 | 12 | 11 | 8 | 8 | 10 | 11 | 1 |
| 14 | 6 | 5 | 9 | 8 | 10 | 9 | 12 |
| 2 | 0 | 0 | 0 | 1 | 8 | 15 | 15 |
| 3 | 7 | 7 | 8 | 10 | 12 | 14 | 10 |

Ukuran gambar 8x8

Skala keabuan = [0,15]

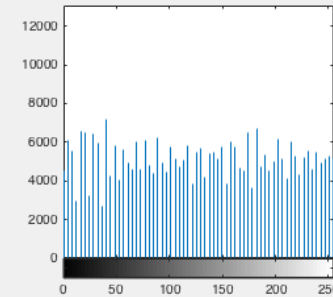
Buat histogram equalization dari citra ukuran 8x8 tersebut!

Histogram Level Keabuan 4 bit

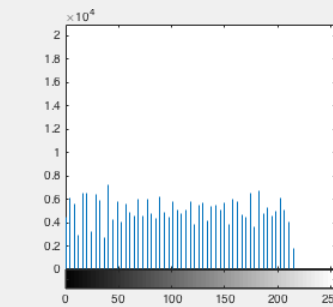


Hasil Histogram Equalization

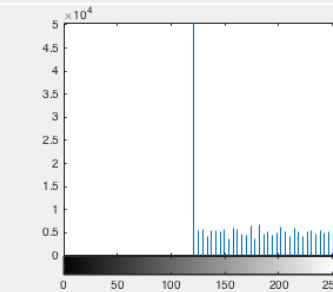
Citra Gelap



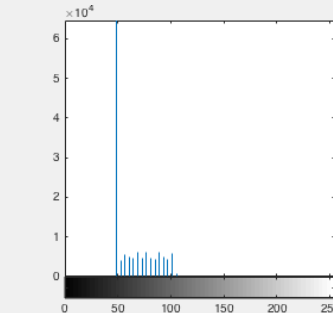
Citra Terang



Kekontrasan Rendah



Kekontrasan Tinggi



- Metode pemrosesan histogram yang sudah dibahas, yaitu **“histogram equalization” bersifat global**, karena piksel-piksel dimodifikasi menggunakan fungsi transformasi berbasis pada intensitas seluruh piksel pada citra.
- **Seringkali diperlukan perbaikan pada suatu daerah yang kecil di dalam citra.**
- Teknik **“histogram equalization”** dapat diterapkan untuk **perbaikan lokal**.

Caranya:

1. Definisikan daerah ketetanggaan (neighborhood)
2. Pada setiap lokasi piksel, histogram dari piksel-piksel dalam neighborhood dihitung.
3. Tentukan fungsi transformasi **“histogram equalization”** dan fungsi ini digunakan untuk memetakan intensitas piksel pada pusat neighborhood.
4. Ulangi langkah 2–3 untuk seluruh piksel dalam citra.

Hasil dari Perbaikan Lokal

Contoh :

terdapat citra dengan 4 level keabuan dan ukuran citra 3x3 sebagai berikut :

$$A = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 1 & 3 \\ 4 & 1 & 4 \end{bmatrix}$$

1. Duplikasi matrix A ke matrix B dengan memberi nilai intensitas 0 di sekeliling matrix A

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 3 & 0 \\ 0 & 4 & 1 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

2. Ambil matrix 3x3 dari matrix B dari posisi pertama (1,1)

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 3 & 0 \\ 0 & 4 & 1 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Kita akan mengganti nilai 2 dengan nilai baru

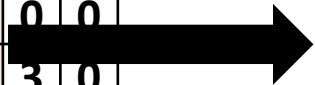
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Hasil dari Perbaikan Lokal

3. Hitung PDF dan CDF dari matrix C

B =

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 2 | 1 | 0 | 0 |
| 0 | 0 | 1 | 3 | 0 |
| 0 | 4 | 1 | 4 | 0 |
| 0 | 0 | 0 | 0 | 0 |



C =

| | | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 2 | 1 |
| 0 | 0 | 1 |

| i | ni | PDF(ni/9) | PDF*ri | PDF | CDF |
|---|----|-----------|--------|-----|-----|
| 0 | 6 | 0.67 | 2.67 | 3 | 3 |
| 1 | 2 | 0.22 | 0.89 | 1 | 4 |
| 2 | 1 | 0.11 | 0.44 | 0 | 4 |
| 3 | 0 | 0.00 | 0.00 | 0 | 4 |
| 4 | 0 | 0.00 | 0.00 | 0 | 4 |

4. Nilai 2 diganti dengan hasil CDF = 4 pada matrik

B =

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 2 | 1 | 0 | 0 |
| 0 | 0 | 1 | 3 | 0 |
| 0 | 4 | 1 | 4 | 0 |
| 0 | 0 | 0 | 0 | 0 |



B =

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 4 | 1 | 0 | 0 |
| 0 | 0 | 1 | 3 | 0 |
| 0 | 4 | 1 | 4 | 0 |
| 0 | 0 | 0 | 0 | 0 |

5. Lakukan perhitungan 3-5 untuk menghitung semua nilai intensitas di matrix A

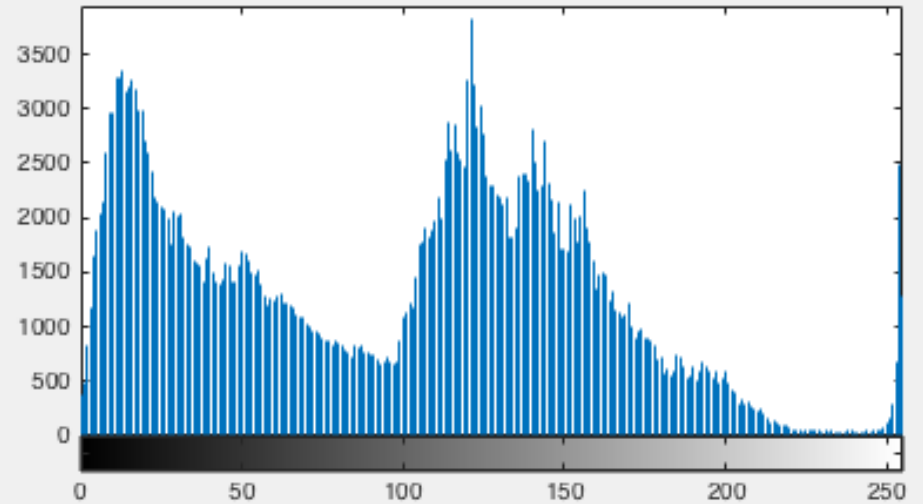
| | | |
|---|---|---|
| 2 | 1 | 0 |
| 0 | 1 | 3 |
| 4 | 1 | 4 |



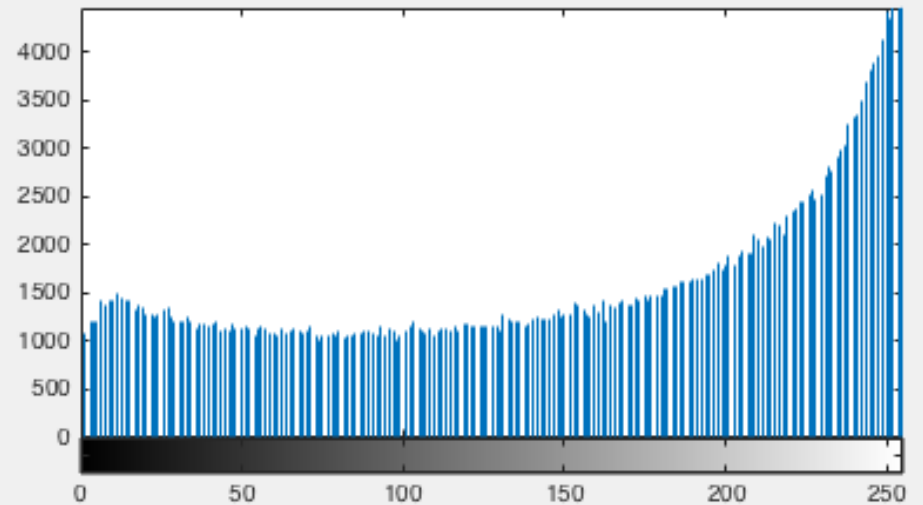
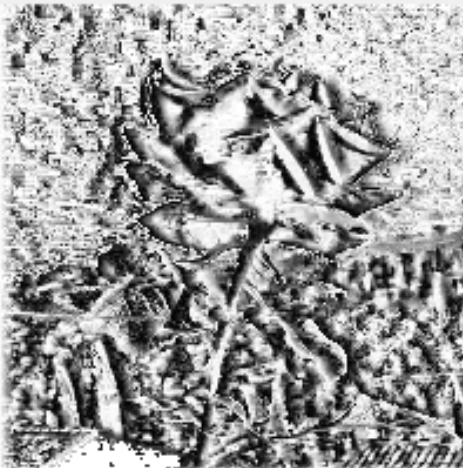
| | | |
|---|---|---|
| 4 | 3 | 3 |
| 2 | 2 | 4 |
| 4 | 3 | 4 |

Hasil dari Perbaikan Lokal

Sebelum Local Histogram Equalization

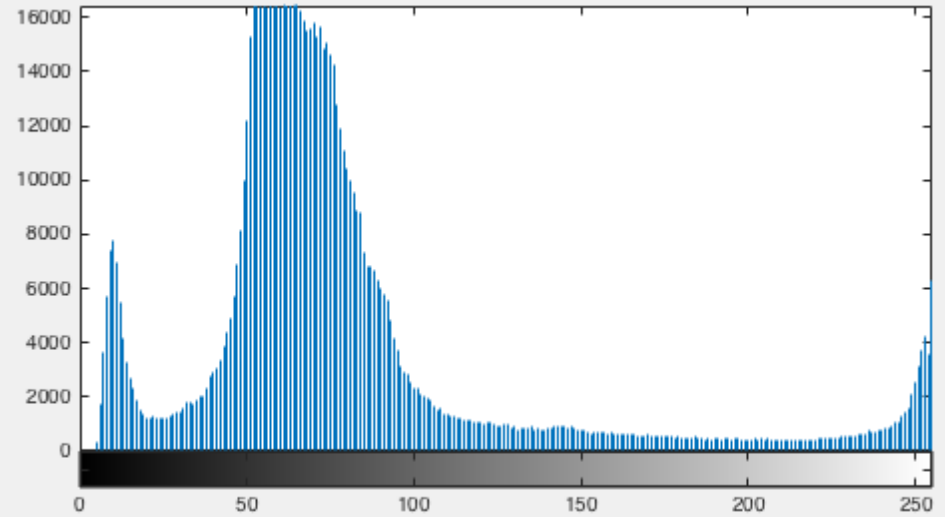


Setelah Local Histogram Equalization

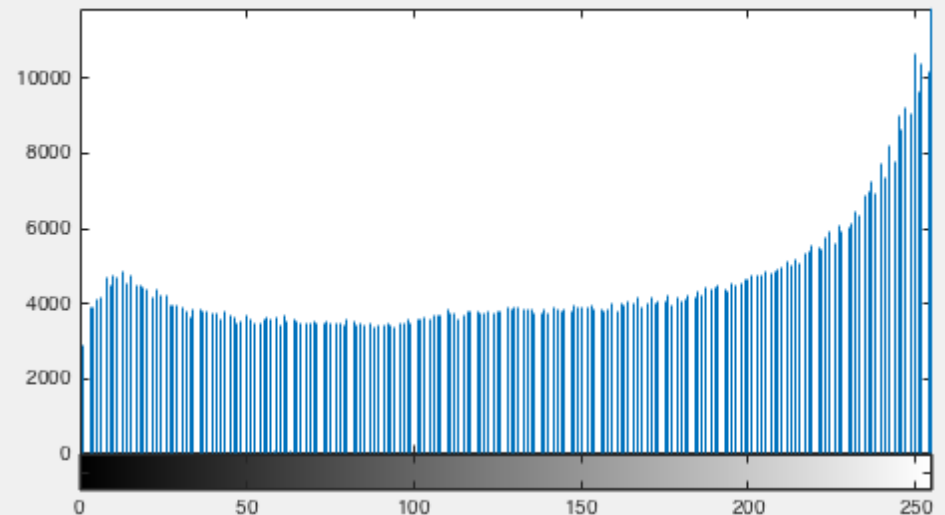


Hasil dari Perbaikan Lokal

Sebelum Local Histogram Equalization



Setelah Local Histogram Equalization



Penggunaan Nilai Statistik dari Histogram Untuk Perbaikan Citra

- Selain menggunakan histogram secara langsung untuk perbaikan citra, dapat pula digunakan parameter-parameter statistik yang didapat dari histogram.
- Parameter statistik yang bisa digunakan adalah:

Mean : yaitu rata-rata tingkat keabuan dalam citra

Variance : yaitu rata-rata kekontrasan citra.

Deviasi standard didefinisikan sebagai akar dari variance

Penggunaan Nilai Statistik dari Histogram Untuk Perbaikan Citra



Misalkan r adalah variabel random diskrit yang menyatakan tingkat keabuan diskrit dalam range $[0, L-1]$, dan $p(r_i)$ adalah komponen “normalized histogram” pada nilai ke- i dari r_i . Bisa diasumsikan bahwa $p(r_i)$ adalah estimasi probabilitas kemunculan tingkat keabuan r_i .

Mean dari r bisa dihitung dengan:

$$m = \sum_{i=0}^{L-1} r_i p(r_i)$$

Variance dari r bisa dihitung dengan: $\sigma^2(r) = \sum_{i=0}^{L-1} (r_i - m)^2 p(r_i)$

Mean dan variance global diukur terhadap seluruh citra dan digunakan untuk menilai intensitas dan kontras citra secara keseluruhan.

Mean dan variance lokal digunakan sebagai dasar untuk membuat perubahan di dalam citra, dimana perubahan tersebut tergantung pada karakteristik di suatu sub daerah di dalam citra.

Mean lokal adalah ukuran tingkat keabuan rata-rata dalam neighborhood dan variance adalah ukuran kontras dalam neighborhood

Contoh : Histogram Statistik

Sebuah citra 2-bit (00, 01, 10, 11) berukuran 5x5 seperti dibawah

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 2 |
| 1 | 2 | 3 | 0 | 1 |
| 3 | 3 | 2 | 2 | 0 |
| 2 | 3 | 1 | 0 | 0 |
| 1 | 1 | 3 | 2 | 2 |

► Citra 2-bit memiliki jumlah nilai $L=4$ dengan nilai 0 sampai dengan 3.

► Total elemen dari citra/jumlah pixel adalah $M \times N = 25$.

$$p(r_0) = \frac{6}{25} = 0.24$$

$$p(r_1) = \frac{7}{25} = 0.28$$

$$p(r_2) = \frac{7}{25} = 0.28$$

$$p(r_3) = \frac{5}{25} = 0.20$$

$$m = \sum_{i=0}^{L-1} r_i p(r_i) = (0)(0.24) + (1)(0.28) + (2)(0.28) + (3)(0.20) = 1.44$$

Latihan : Hitung variansi dan standar deviasinya!

Salah satu Contoh Perbaikan Citra dengan Pendekatan Statistik : Filtering



Filtering kita bahas lebih lanjut minggu depan ya 😊

Thank You....