

Pengolahan Citra Digital

DASAR-DASAR PENGENALAN CITRA
IFK15036, 3 credits



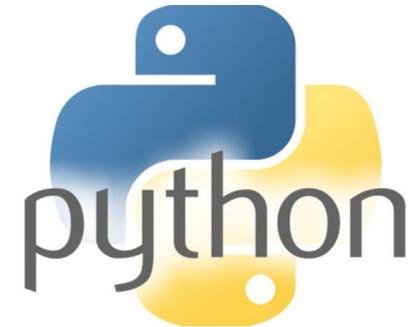
Team Teaching :

1. Dr. Fitri Utamingrum, S.T, M.T

OUTLINE BAB-2

- Representasi Citra Digital
- Kuantisasi citra
- Kualitas citra
- Cara membaca citra, mendapatkan ukuran citra, menampilkan citra
- Mengenal jenis citra (citra berwarna, citra berskala keabuan, citra biner)
- Mengkonversi jenis citra
- Histogram Citra
- Konsep operasi ketetanggaan piksel
- Operasi Aritmatika pada dua buah citra
 - Penjumlahan
 - Pengurangan
 - Perkalian
- Operasi Geometrik (translasi, rotasi, dilatasi, pencerminan)

TOOLS



- Bisa menggunakan salah satu *tools* → Java, Matlab, GNU Octave, atau Python. Namun **di perkuliahan ini contoh-contoh yang dibuat akan menggunakan MATLAB**
- Meski di beberapa tools sudah memiliki *library* atau *package* sendiri untuk pengolahan citra digital, kita harus memahami proses di dalamnya, sehingga pada beberapa bagian materi perkuliahan ini kita akan tetap belajar untuk *building from scratch*.

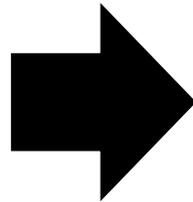
Ingat ya... PLAGIARISME == 0

Representasi Citra

Ruang warna : RGB



Citra berukuran 58x58



100	100	101	101	99	99	100	99	101	103	102	102	103	105	104	104
100	102	102	102	101	102	103	104	102	104	104	104	104	106	105	107
103	104	104	104	104	104	105	104	105	107	106	105	106	106	106	109
106	105	105	105	105	106	107	106	108	108	107	107	108	108	109	110
107	107	107	106	107	108	108	109	109	109	108	109	109	110	110	112
107	108	108	107	109	110	110	111	111	110	111	111	111	112	112	114
109	108	108	110	111	110	111	112	114	112	113	113	114	114	114	114
110	109	108	110	112	112	112	114	116	114	115	114	115	115	115	116
110	111	111	112	113	113	114	116	117	115	116	116	116	117	118	119
113	114	114	114	114	115	115	116	117	118	118	118	118	119	120	120
114	114	115	116	117	117	117	118	119	120	120	120	120	120	120	121
116	116	117	118	120	119	120	120	120	121	122	122	121	122	122	123
117	118	118	120	121	121	121	120	121	121	123	123	123	124	124	123
119	120	120	121	122	122	123	123	123	123	124	123	124	125	124	123
121	121	121	123	122	124	125	125	125	125	125	125	124	125	122	124
123	123	123	124	124	124	125	127	126	127	128	128	128	128	127	79
123	124	124	126	127	127	127	130	129	129	130	130	131	131	131	83
124	125	125	127	128	129	129	131	131	130	131	132	131	131	131	129
126	127	129	129	130	130	131	132	132	132	132	133	133	132	129	134
126	128	130	132	132	132	132	133	134	135	136	136	136	133	134	110
129	130	130	132	134	135	136	136	136	136	137	137	135	135	140	89
132	133	131	134	137	136	137	137	138	138	140	139	138	139	137	137
133	133	134	137	138	137	138	139	139	141	143	141	141	142	139	141
134	134	138	139	139	139	140	142	142	143	145	144	144	143	141	143

Data penyusun citra 58x58 pada *channel R*

Digitalisasi Citra

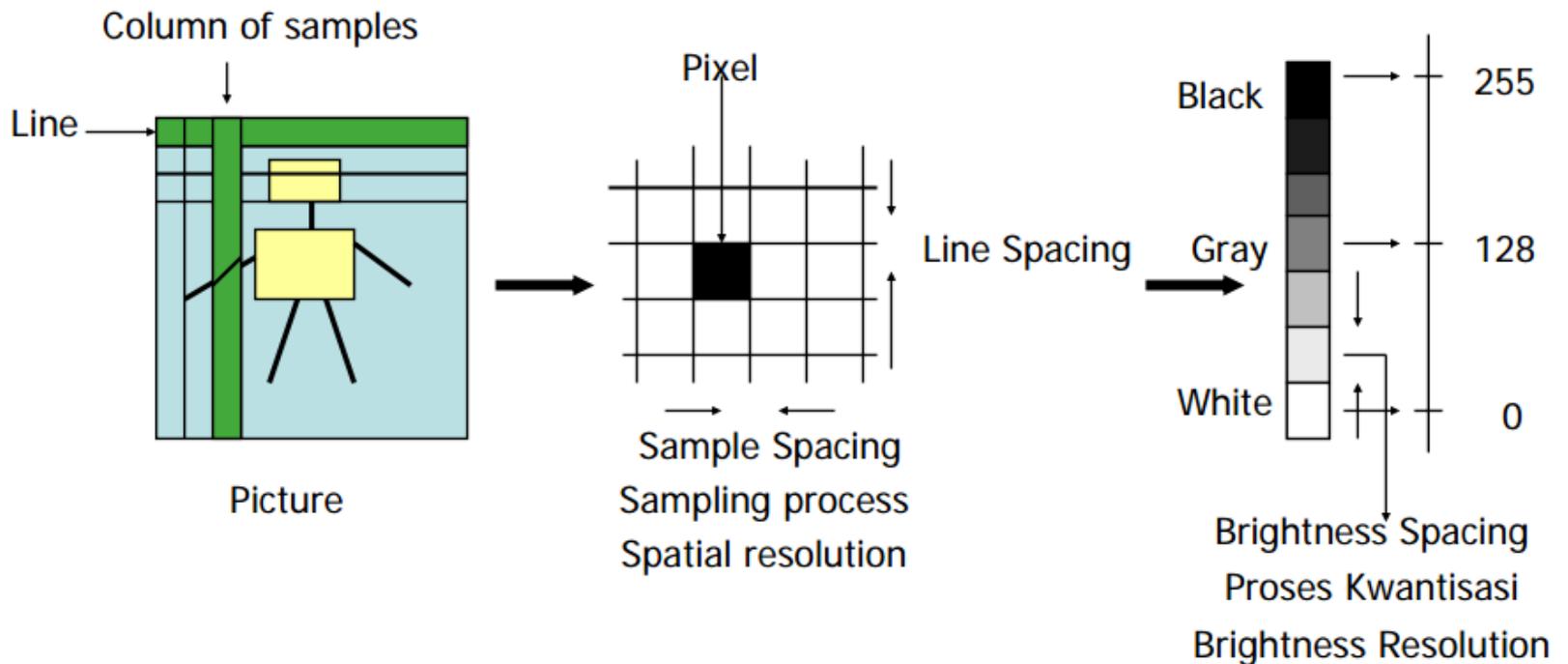
Sampling

- Proses pengambilan nilai diskrit pada koordinat ruang (x,y) melalui *grid*

Kuantisasi

- Proses pengelompokan nilai tingkat keabuan citra kontinu ke dalam beberapa level.
- Atau merupakan proses membagi skala keabuan $(0,L)$ menjadi G buah level yang dinyatakan dengan bilangan bulat integer

Digitalisasi Citra



Sumber: Dimodifikasi dari Castleman, 1996

Contoh Sampling Citra



Original Image
220 x 220



Resize = 0.3
66 x 66



Resize = 0.1
22 x 22

Kualitas Citra

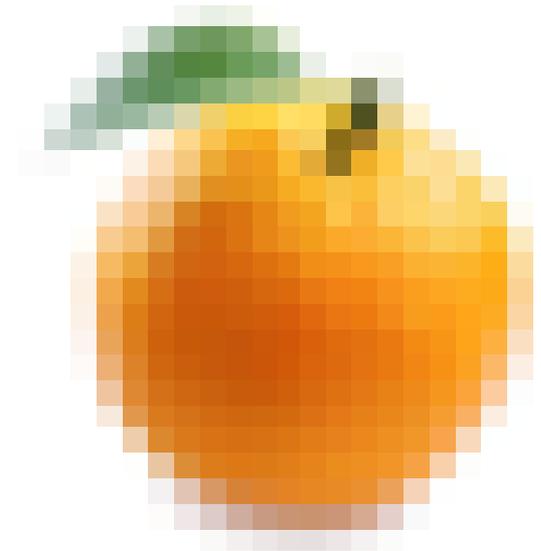
Kualitas citra dipengaruhi juga oleh jumlah piksel yang menyusun suatu citra (resolusi citra).



Original Image
220 x 220



Resize = 0.3
220 x 66



Resize = 0.1
22 x 22

Efek resolusi berdasar jumlah piksel pada citra ketika gambar **disajikan dengan ukuran yang sama** bisa mempengaruhi kualitasnya.

Kuantisasi Citra

- Citra digital dibentuk melalui pendekatan yang dinamakan kuantisasi.
- Kuantisasi adalah proses diskritisasi dari bentuk kontinu.

Tabel 1. Jangkauan nilai pada citra keabuan

Bit per piksel (<i>Pixel depth</i>)	Derajat keabuan ($G=2^m$) dan jangkauan	Penggunaan
1	$2^1 = 2$ (0-1)	Citra biner:dokumen faksimili
2	$2^2 = 4$ (0-3)	
5	$2^5 = 32$ (0-31)	
8	$2^8 = 256$ (0-255)	Foto
12	$2^{12}=4096$ (0-4095)	Foto kualitas tinggi
14	$2^{14}=16384$ (0-16383)	Foto kualitas profesional
16	$2^{16}=65536$ (0-65535)	Citra kualitas tertinggi : citra kedokteran dan astronomi

Kuantisasi Citra

Dalam pengolahan citra, kuantisasi dari level keabuan menentukan kecermatan hasilnya.



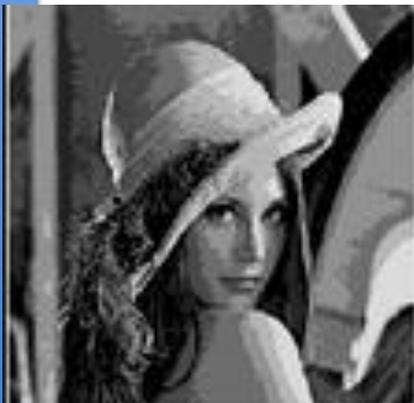
N=6
4



N=3
2



N=1
6



N=8



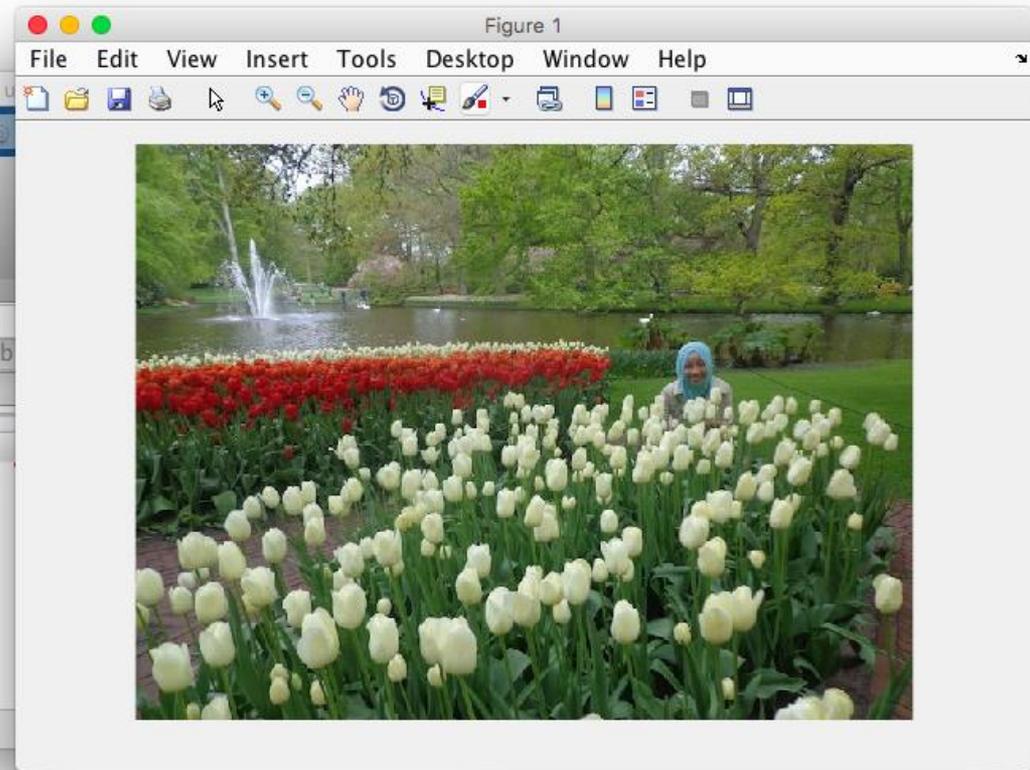
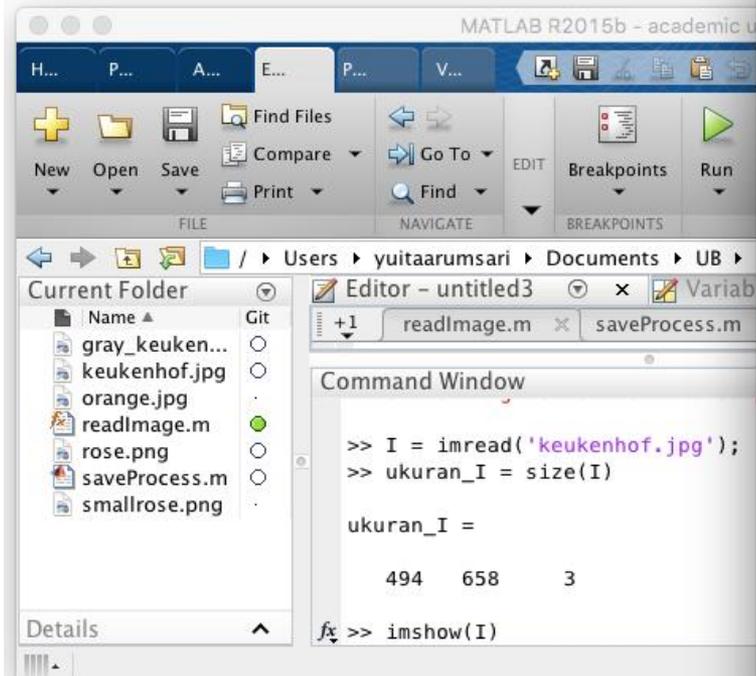
N=4



N=2

Membaca, mendapatkan ukuran citra, dan menampilkan citra

```
>> I = imread('keukenhof.jpg'); % membaca citra
>> ukuran_I = size(I) %mendapatkan ukuran citra
ukuran_I =
    494    658     3
>> imshow(I) %menampilkan citra
```

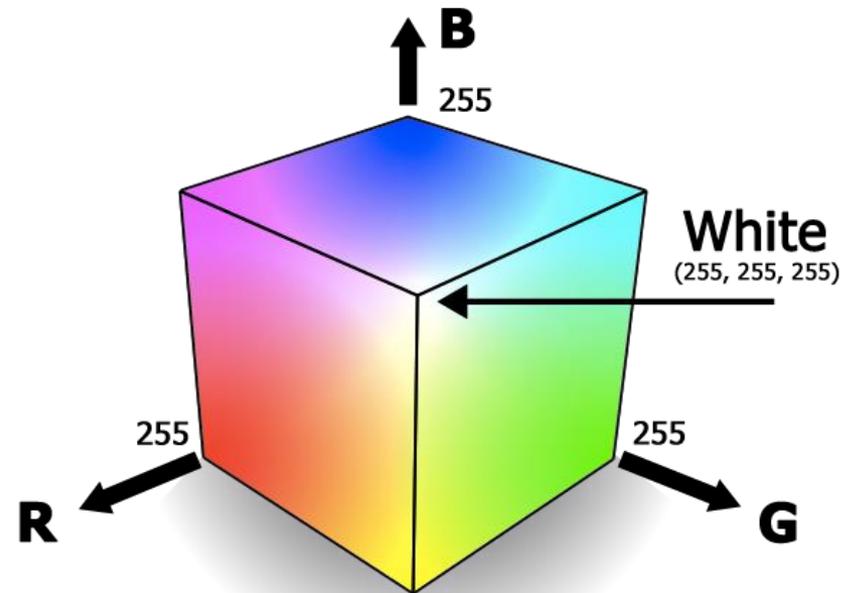


Citra Berwarna (RGB)

- ✓ Secara *default* kamera akan menghasilkan gambar yang mempunyai ruang warna RGB yang artinya *Red*, *Green*, dan *Blue*
- ✓ Setiap komponen warna pada RGB mengandung 8 bit yang nilainya berkisar antara 0-255
- ✓ Kemungkinan pilihan warna yang dapat disajikan adalah $255 \times 255 \times 255 = 16.581.375$ warna

Contoh warna dan nilai penyusun warna

Warna	R	G	B
Merah	255	0	0
Hijau	0	255	0
Biru	0	0	255
Hitam	0	0	0
Putih	255	255	255
Kuning	0	255	255



Cara mendapatkan komponen R, G, dan B

```
R = I(:,:,1); %mendapatkan komponen R  
G = I(:,:,2); %mendapatkan komponen G  
B = I(:,:,3); %mendapatkan komponen B  
Subplot(1,2,1),imshow(R), title('R')  
Subplot(1,2,2),imshow(R), title('G')  
Subplot(1,2,3),imshow(R), title('B')
```

R



G



B



CITRA BERSKALA KEABUAN

- ◆ Menghasilkan gambar abu-abu dimana warna yang dihasilkan dinyatakan dengan **intensitas**.
- ◆ Nilai intensitas berkisar antara 0-255



CITRA BINER

- ◆ Setiap piksel hanya dinyatakan dalam dua kemungkinan nilai yaitu 0 dan 1.
- ◆ Nilai 0 menyatakan warna hitam dan nilai 1 menyatakan warna putih



Konversi Jenis Citra : Citra Berwarna ke Citra Keabuan

```
%diasumsikan bahwa salah satu contoh  
% persamaan yang biasa dipakai untuk melakukan  
% konversi citra berwarna ke citra keabuan adalah  
%  $I = 0.2989 * R + 0.5870 * G + 0.1141 * B$ 
```

```
rose = imread('rose.png');  
abu2 = uint8(0.2989 * double(rose(:,:,1)) + ...  
            0.5870 * double(rose(:,:,2)) + ...  
            0.1141 * double(rose(:,:,3)));  
imshow(abu2)
```

OUTPUT



Konversi Jenis Citra : Citra Keabuan ke Citra Biner

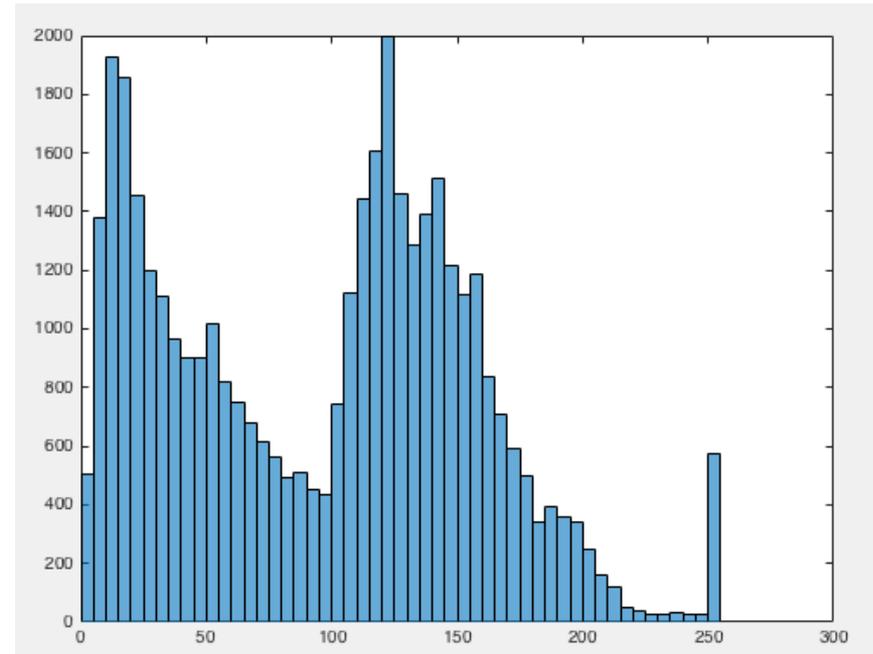
```
% Mengubah dari citra keabuan kedalam citra biner
[a,b] = size(abu2);
threshold = 100 ; % nilai ini bisa berubah
binary = zeros(a,b); % inisialisasi
for baris=1:a
    for kolom=1:b
        if abu2(baris,kolom) >= threshold
            binary(baris,kolom) = 1;
        else
            binary(baris,kolom) = 0;
        end
    end
end
imshow(binary)
```

OUTPUT



Histogram Citra

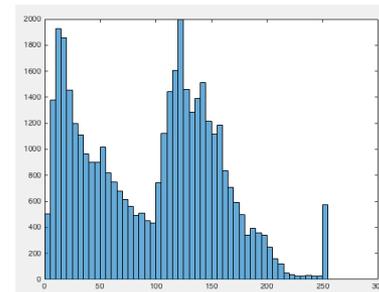
- Histogram citra merupakan diagram yang menggambarkan frekuensi setiap nilai intensitas yang muncul di seluruh piksel citra.
- Histogram digunakan untuk menunjukkan level keabuan yang nilainya dimulai dari 0 hingga 255



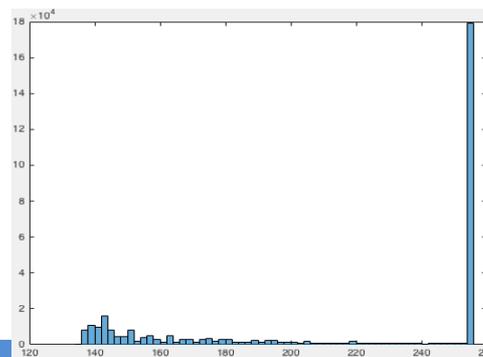
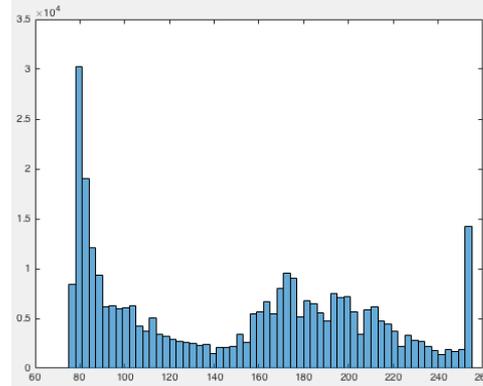
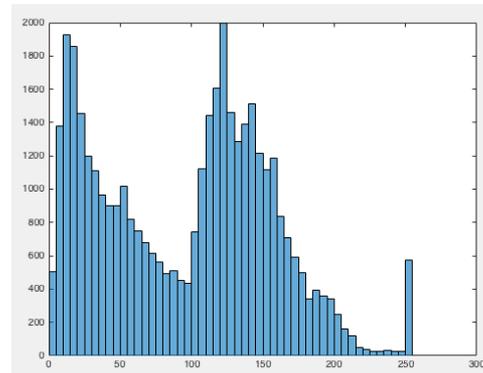
Histogram Citra

- Pada pengolahan citra, histogram mempunyai peran yang cukup penting.
- Manfaat histogram citra :
 1. Berguna untuk mengamati penyebaran intensitas warna dan dapat dipakai untuk pengambilan keputusan, misalnya dalam peningkatan kecerahan atau peregangan kontras serta sebaran warna.
 2. Berguna untuk penentuan batas-batas dalam pemisahan objek dari latar belakangnya.
 3. Memberikan persentase komposisi warna dan tekstur intensitas untuk kepentingan identifikasi citra.

Histogram tidak bisa digunakan untuk mengenali bentuk objek dalam sebuah citra.



Histogram Citra - Brightness

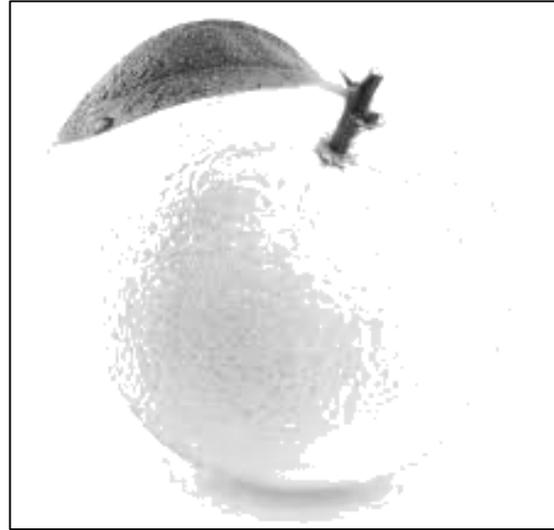
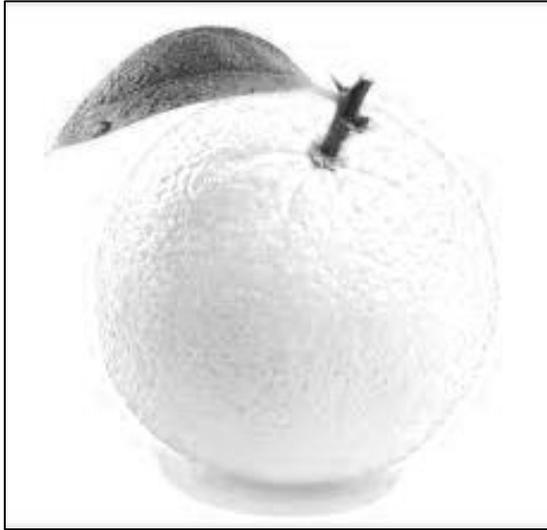


Cara meningkatkan kecerahan pada gambar :

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

```
rose1 =  
imread('grey_rose.png');  
rose1_ = rose1 + 85;  
imshow(rose1_  
histogram(rose1_)
```

Pemotongan Intensitas Citra dari Level Keabuan 0-255



$f1=10 \ f2=230$



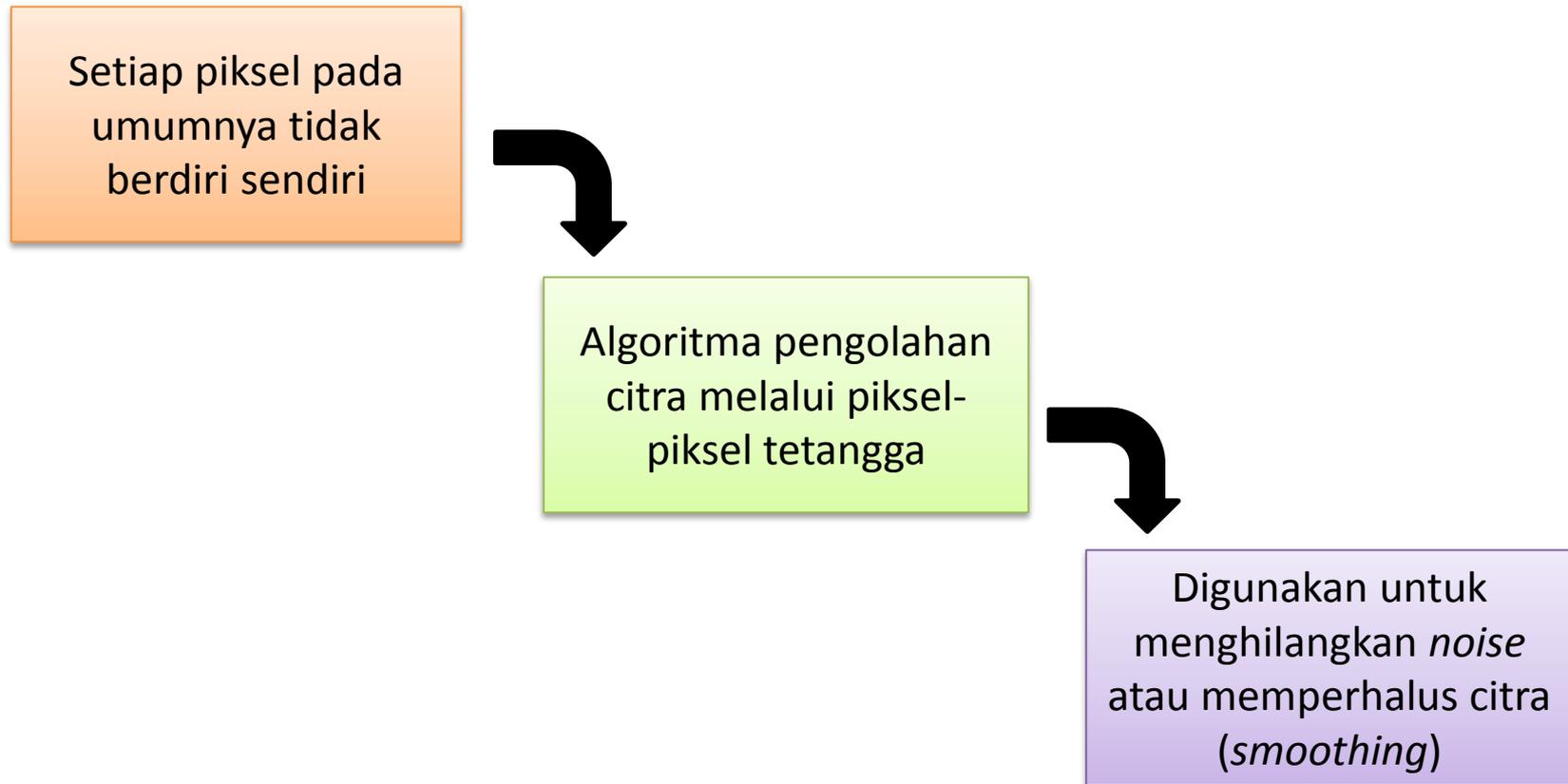
$f1=50 \ f2=150$

Gambar diatas menunjukkan pemotongan level keabuan pada sebuah citra. Dengan konsep histogram citra, maka kita bisa menggunakan aturan nilai intensitas yang $\leq f1$ dan nilai intensitas yang $\geq f2$ diganti dengan 0.

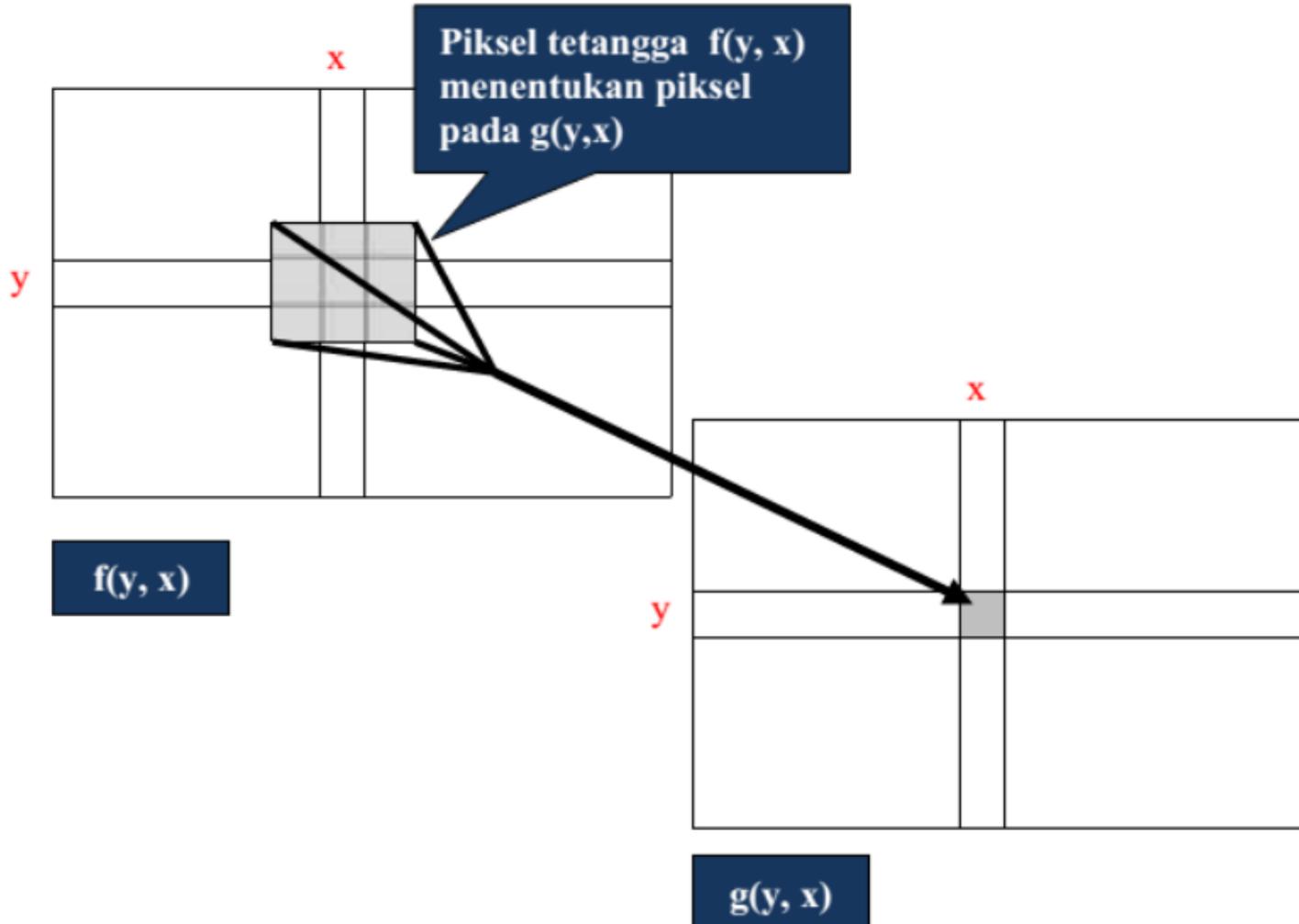
$$g(x, y) \begin{cases} 0, & x \leq f1 \\ f(y, x), & f1 < f(y, x) < f2 \\ 255, & x \geq f2 \end{cases}$$

Operasi Ketetangaan Piksel

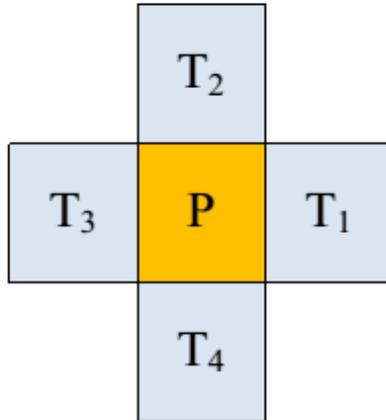
- Operasi ketetangaan piksel adalah pengolahan citra untuk mendapatkan nilai suatu piksel yang melibatkan piksel-piksel tetangganya.



Operasi Ketetanggaan Pixel

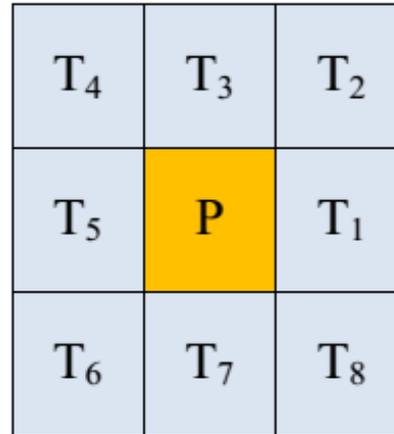


Jenis Ketetanggaan



4 ketetanggaan

$$\begin{aligned}T_1 &= (b, k+1) \\ T_2 &= (b-1, k) \\ T_3 &= (b, k-1) \\ T_4 &= (b+1, k)\end{aligned}$$



8 ketetanggaan

$$\begin{aligned}T_1 &= (b, k+1) \\ T_2 &= (b-1, k-1) \\ T_3 &= (b, k-1) \\ T_4 &= (b-1, k-1) \\ T_5 &= (b, k-1) \\ T_6 &= (b+1, k-1) \\ T_7 &= (b+1, k-1) \\ T_8 &= (b+1, k+1)\end{aligned}$$

Kita pelajari lebih detail pada pertemuan ke-4

Aritmatika : Operasi Penjumlahan

- ❖ Persamaan yang digunakan:

$$C(x,y)=A(x,y)+B(x,y)$$

- ❖ Dengan $C(x,y)$ adalah citra baru yang setiap pikselnya adalah jumlah dari intensitas tiap piksel pada A dan B
- ❖ Jika hasil penjumlahan lebih besar dari 255, maka intensitas dapat dibulatkan menjadi 255
- ❖ Sering digunakan untuk penggabungan dua buah citra dan *watermarking* tampak (*visible watermarking*)

orange



rose



orange+rose



Aritmatika : Operasi Pengurangan

- ❖ Persamaan yang digunakan:

$$C(x,y)=A(x,y)-B(x,y)$$

- ❖ Dengan piksel citra C adalah hasil pengurangan intensitas piksel citra A dengan citra B
- ❖ Ada kemungkinan hasil pengurangan menghasilkan nilai negatif → diperlukan proses *clipping /cropping*

Contoh aplikasi:

- Untuk mendeteksi pergerakan obyek
- Untuk mendeteksi keamanan pada suatu ruangan

Pengurangan citra seringkali digunakan untuk mendeteksi perubahan obyek dalam selang waktu tertentu

orange



rose



orange-rose



Aritmatika : Operasi Perkalian

- Persamaan yang digunakan adalah

$$C(x,y)=A(x,y)B(x,y)$$

- Perkalian citra sering digunakan untuk mengoreksi ketidaklinearan sensor dengan mengalikan matriks citra dengan matriks koreksi

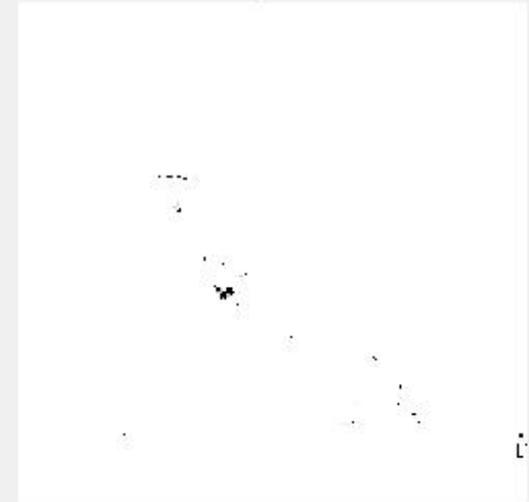
orange



rose



orange*rose



- Koordinat piksel berubah akibat transformasi, sedangkan intensitasnya tetap
- Contoh:
 - Translasi (Pergeseran)
 - Rotasi
 - Penyekalaan (Dilatasi)
 - Pencerminkan

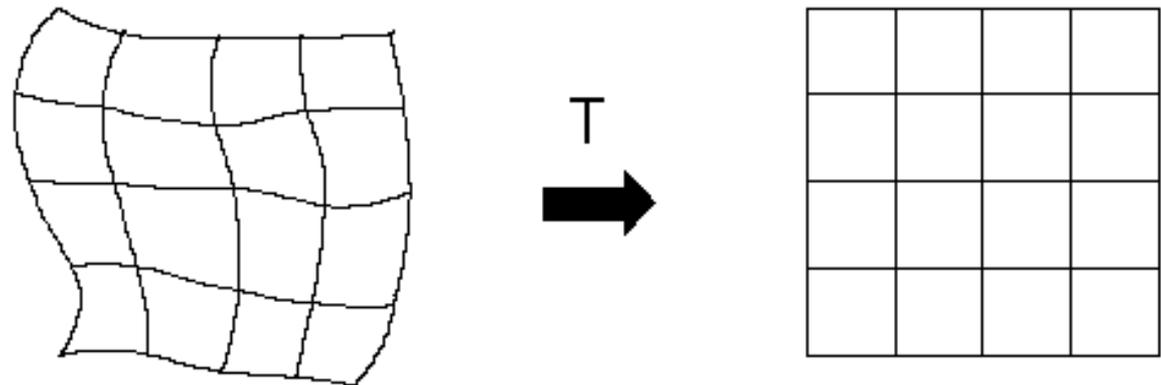


Figure 4.5 Geometric transform on a plane.

Translasi

- Translasi dilakukan berdasar rumus:
 $x' = x + m$
 $y' = y + n$
- m adalah besarnya pergeseran dalam arah x , sedangkan n adalah besarnya pergeseran dalam arah y
- Jika citra semula adalah A , dan citra hasil translasi adalah B , maka translasi dapat dilakukan dengan cara:

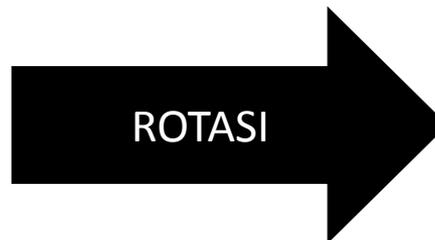
$$B(x, y) = A(x + m, y + n)$$



ROTASI

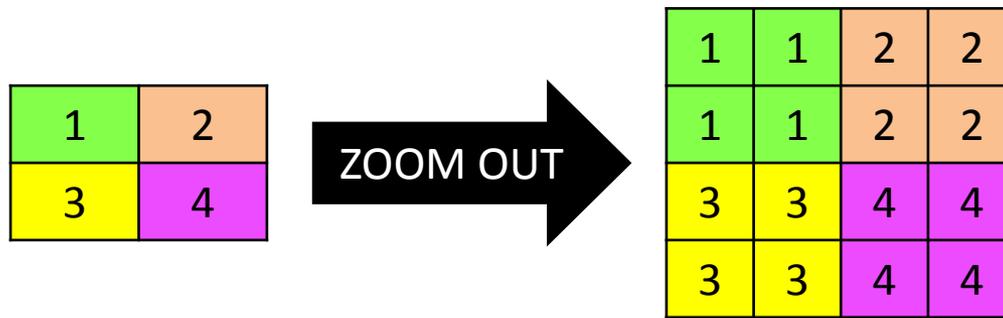
- Rotasi dilakukan dengan persamaan:
 $x' = x \cos(\theta) - y \sin(\theta)$
 $y' = x \sin(\theta) + y \cos(\theta)$
- Dalam hal ini, θ adalah sudut rotasi berlawanan dengan arah jarum jam
- Jika citra semula adalah A, dan citra hasil rotasi adalah B, maka rotasi citra dari A ke B:

$$A(x,y) = B(x \cos(\theta) - y \sin(\theta), x \sin(\theta) + y \cos(\theta))$$



- Penskalaan citra/image zooming: pengubahan ukuran citra (pembesaran/zoom out atau pengecilan /zoom in)
- Rumus penskalaan citra:
 $x' = s_x \cdot x$
 $y' = s_y \cdot y$
- s_x dan s_y adalah faktor penskalaan, masing-masing dalam arah x dan y
- Jika citra semula adalah A dan citra hasil penskalaan adalah B, maka penskalaan citra dinyatakan sebagai:

$$B(x', y') = B(s_x \cdot x, s_y \cdot y) = A(x, y)$$



PENCERMINAN

- Adalah operasi geometri yang sama dengan pencerminan
- Dua macam pencerminan:
 - Horizontal
 - Adalah pencerminan terhadap sumbu Y
 - $B(x,y)=A(N-x,y)$
 - Vertikal
 - Adalah pencerminan terhadap sumbu X
 - $B(x,y)=A(x,M-y)$



Citra I



Citra Fo

Urutan baris untuk citra Fo = baris citra I

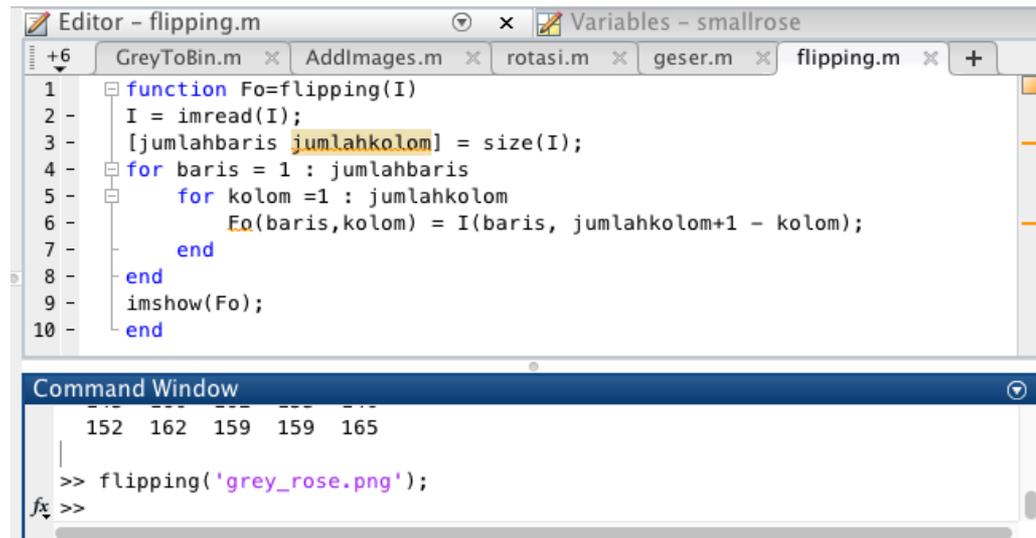
Kolom pertama citra Fo = Kolom terakhir citra I

Contoh Fungsi Pencermianan pada Matlab ditunjukkan sebagai berikut :

```
function Fo=flipping(I)
I = imread(I);
[jumlahbaris jumlahkolom] = size(I);
for baris = 1 : jumlahbaris
    for kolom =1 : jumlahkolom
        Fo(baris,kolom) = I(baris, jumlahkolom+1 - kolom);
    end
end
imshow(Fo);
end
```

Fungsi tersebut dijalankan dengan perintah :

```
>> flipping('grey_rose.png');
```



The screenshot shows the MATLAB Editor window with the following code in the 'flipping.m' file:

```
1 function Fo=flipping(I)
2 I = imread(I);
3 [jumlahbaris jumlahkolom] = size(I);
4 for baris = 1 : jumlahbaris
5     for kolom =1 : jumlahkolom
6         Fo(baris,kolom) = I(baris, jumlahkolom+1 - kolom);
7     end
8 end
9 imshow(Fo);
10 end
```

The Command Window shows the execution of the function:

```
>> flipping('grey_rose.png');
```

Referensi



- Rafael C. Gonzalez. 2002. Digital Image Processing 2nd Edition. Prentice Hall. Upper Saddle River, New Jersey 07458.
- Abdul Kadir dan Adhi Susanto. 2013. Teori dan Aplikasi Pengolahan Citra. Penrbit Andi, Yogyakarta.

Thank You....

TUGAS

Pilih satu bahasa pemrograman yang paling kelompok kalian kuasai diantara Java, Matlab, Octave atau Python. Karena tidak akan diijinkan untuk ganti bahasa pemrograman untuk mengerjakan tugas berikutnya sampai akhir perkuliahan semester ini.

- Membuat program untuk menerapkan operasi geometri :
 - ✓ Translasi
 - ✓ Rotasi
 - ✓ Dilatasi → Zoom in
 - ✓ Dilatasi → Zoom out
- Laporan/Dokumentasi berisi
 - ✓ Logika dan penjelasan perhitungan manual.
 - ✓ *Screenshot* hasil program yang telah dibuat

- Tugas dikerjakan kelompok dimana pembentukan kelompok dilakukan oleh Dosen (mahasiswa tidak bisa memilih sendiri anggota kelompoknya).
- Buat account pada <https://github.com/> untuk mengupload tugas yang berupa coding. Tentang Git dan Github dipelajari sendiri.