Computer Vision

DASAR-DASAR PENGENALAN CITRA

Prof. Dr. Eng. Fitri Utaminingrum, ST, MT

OUTLINE

- Tool
- Matrik citra digital
- Citra biner
- Citra 4 bit (Gray dan color)
- Citra 8 bit (Gray dan Color)
- Citra berwarna 16 bit
- Citra Berwarna 24 bit

TOOLS

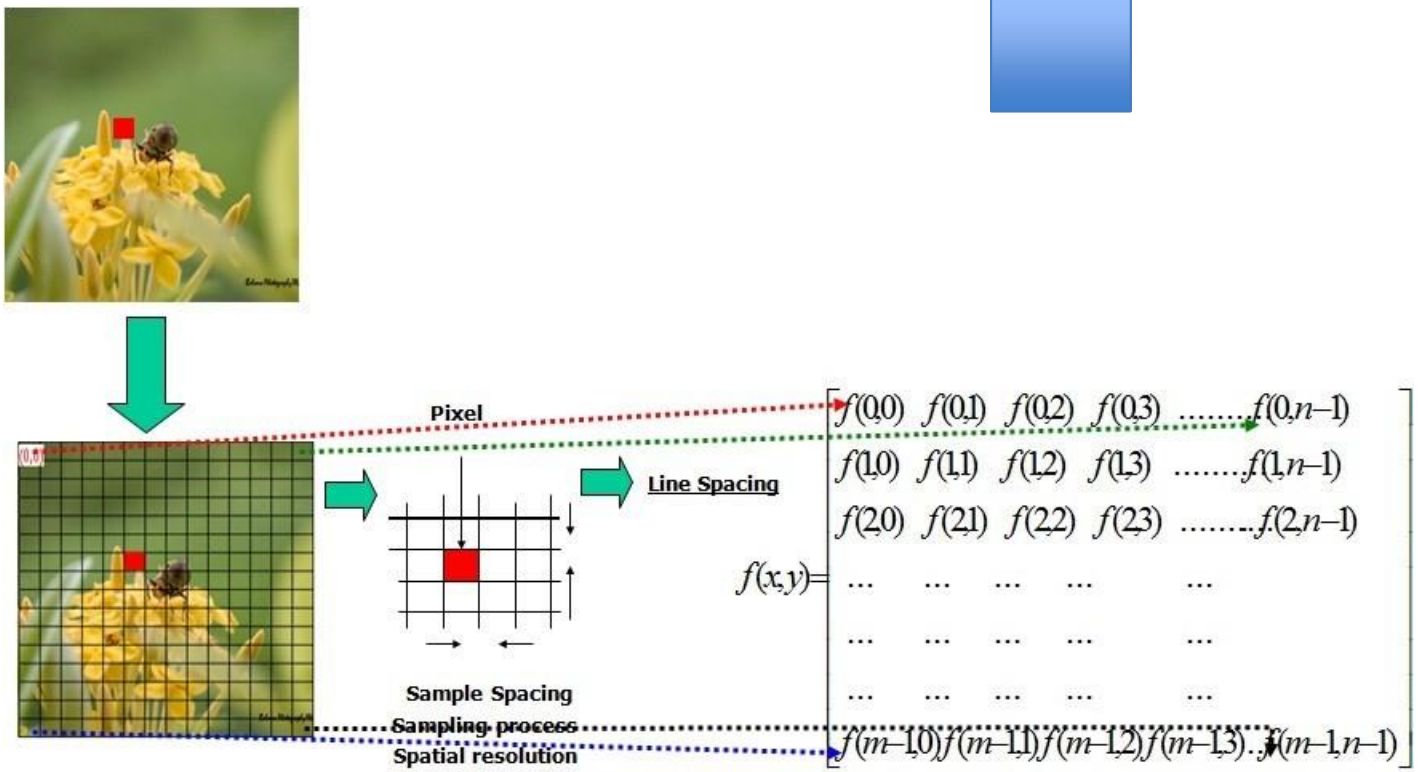
- Bisa menggunakan salah satu *tools* → Java, Matlab, Python dll. Namun di perkuliahan ini **contoh-contoh yang dibuat akan menggunakan MATLAB dan Python**
- Meski di beberapa tools sudah memiliki *library* atau *package* sendiri untuk pengolahan citra digital, kita harus memahami proses di dalamnya,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

TOOLS

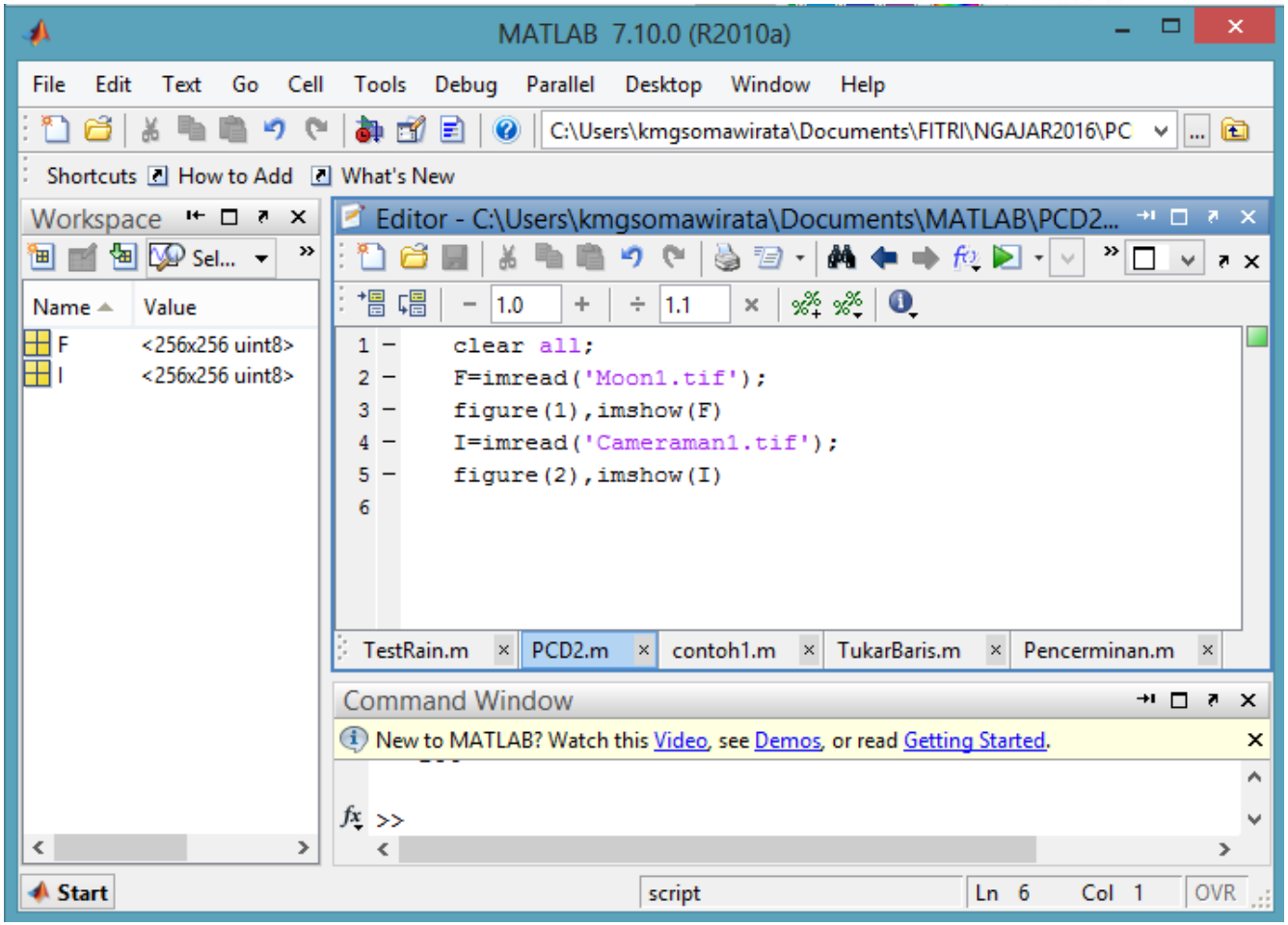
- Minimal Matlab versi 2010 atau terbaru

- Opencv -2.4.11
- Python -3.7.1
- numpy-1.9.1-win32-superpack-python2.7
- pycharm-community-2017.2.2
- Atau versi terbaru

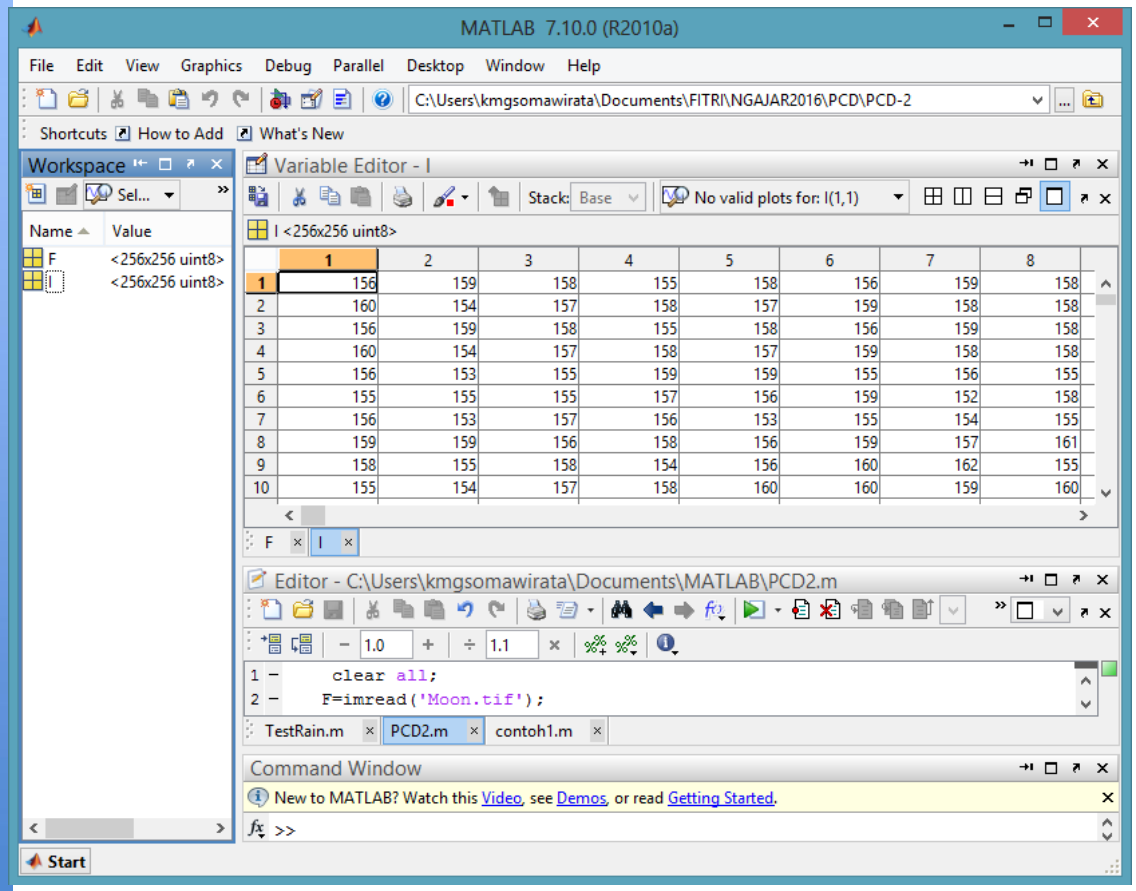
MATRIK CITRA



MATRIK CITRA



MATRIK CITRA



	1	2	3	4	5	6	7	8
1	156	159	158	155	158	156	159	158
2	160	154	157	158	157	159	158	158
3	156	159	158	155	158	156	159	158
4	160	154	157	158	157	159	158	158
5	156	153	155	159	159	155	156	155
6	155	155	155	157	156	159	152	158
7	156	153	157	156	153	155	154	155
8	159	159	156	158	156	159	157	161
9	158	155	158	154	156	160	162	155
10	155	154	157	158	160	160	159	160

$$f(x,y) = \begin{vmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,M} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,M} \\ a_{3,1} & a_{3,2} & a_{3,3} & \cdots & a_{3,M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{N,1} & a_{N,2} & a_{N,3} & \cdots & a_{N,M} \end{vmatrix}$$

CONTOH DALAM PYTHON

Membaca file gambar dan menampilkannya

KOMPUTASICITRA - [C:\Users\Fitri Utaminingrum\PycharmProjects\KOMPU...

File Edit View Navigate Code Refactor Run Tools VCS Window Help

KOMPUTASICITRA > Coba1.py > Coba1

Coba1.py

KOMPUTASICITRA C:\Use

br30.jpg
br50.jpg
br70.jpg
br80.jpg
br90.jpg
br100.jpg
br110.jpg
br120.jpg
br130.jpg

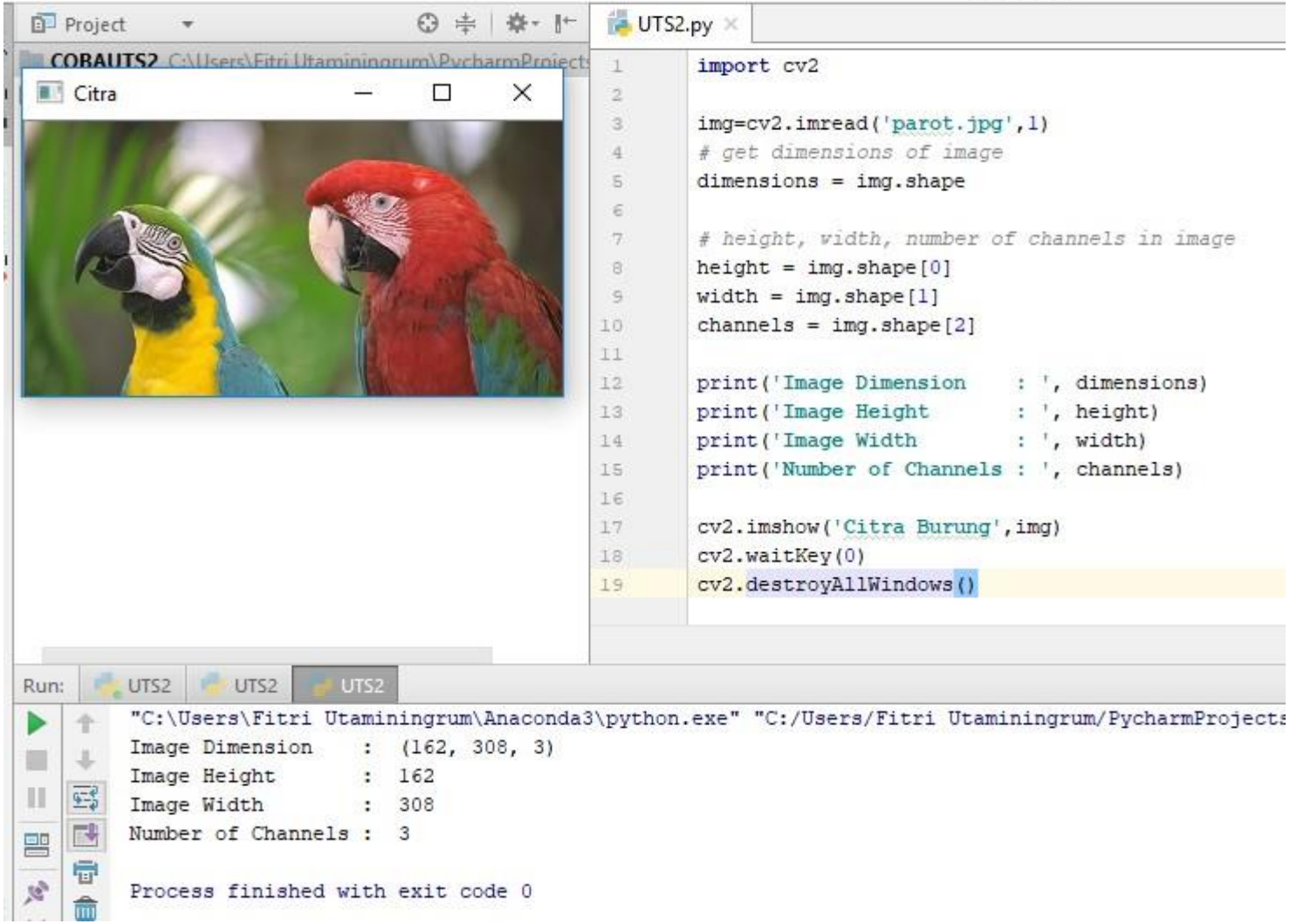
```
1  import numpy as up
2  import cv2
3  frame = cv2.imread('br110.jpg',1)
4  cv2.imshow('original',frame)
5  cv2.waitKey(0)
6  cv2.destroyAllWindows()
7
```

Run Coba1

```
C:\Python27\python.exe "C:/Users/Fitri Utaminingrum/PycharmProje

Process finished with exit code 0
```

IDE and Plugin Updates: PyCharm Co... (today 6:02 AM) 7:1 n/a UTF-8

CONTOH DIMENSI CITRA



KONVERSI CITRA RGB TO GRAYSCALE

Konversi ke Gray Scale

Setiap pixel mempunyai nilai red (*r*), green (*g*) dan blue (*b*) dengan nilai masing-masing 0-255

Setiap pixel mempunyai nilai derajat keabuan x dengan nilai 0-255

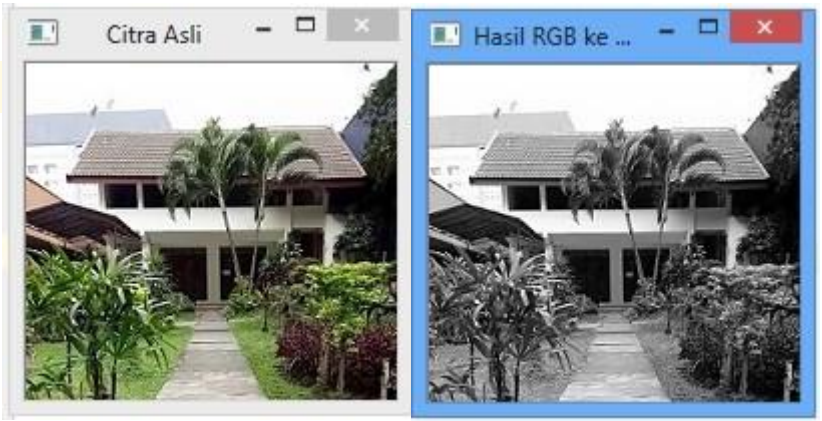
$$x = \frac{r + g + b}{3}$$

$$x = a_r.r + a_g.g + a_b.b$$

dimana : $a_r + a_g + a_b = 1$

KONVERSI CITRA RGB TO GRAYSCALE (PHYTON)

```
1   import cv2
2   import numpy as np
3
4   img = cv2.imread('Gbr1.jpg',1)
5
6   RGBkeGray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
7
8
9   cv2.imshow('Citra Asli',img)
10  cv2.imshow('Hasil RGB ke GRAY',RGBkeGray)
11
12  cv2.waitKey(0)
13  cv2.destroyAllWindows()
14
```



KONVERSI GRAYSCALE TO BINER

Konversi ke Biner

Setiap pixel mempunyai nilai derajat keabuan x dengan nilai 0-255

Setiap pixel mempunyai nilai warna x_{bw} dengan nilai 0 dan 1

$$x_{bw} = \begin{cases} 1 & \text{jika } x \geq 128 \\ 0 & \text{jika } x < 128 \end{cases}$$

$$x_{bw} = \begin{cases} 1 & \text{jika } x \geq \bar{x} \\ 0 & \text{jika } x < \bar{x} \end{cases}$$

Merubah citra Gray ke Biner

- Untuk merubah citra gray dapat dilakukan dengan menggunakan ambang batas/Threshold

- Menggunakan fungsi `cv2.threshold()`

Merubah citra Gray ke Biner

```
1   import cv2
2   import numpy as np
3
4   img = cv2.imread('Gbr1.jpg',1)
5
6   RGBkeGray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
7   ret,thresh1 = cv2.threshold(RGBkeGray,127,255,cv2.THRESH_BINARY)
8
9
10  cv2.imshow('Citra Asli',img)
11  cv2.imshow('Hasil RGB ke GRAY',RGBkeGray)
12  cv2.imshow('Hasil GRAY ke BINER',thresh1)
13
14  cv2.waitKey(0)
15  cv2.destroyAllWindows()
```

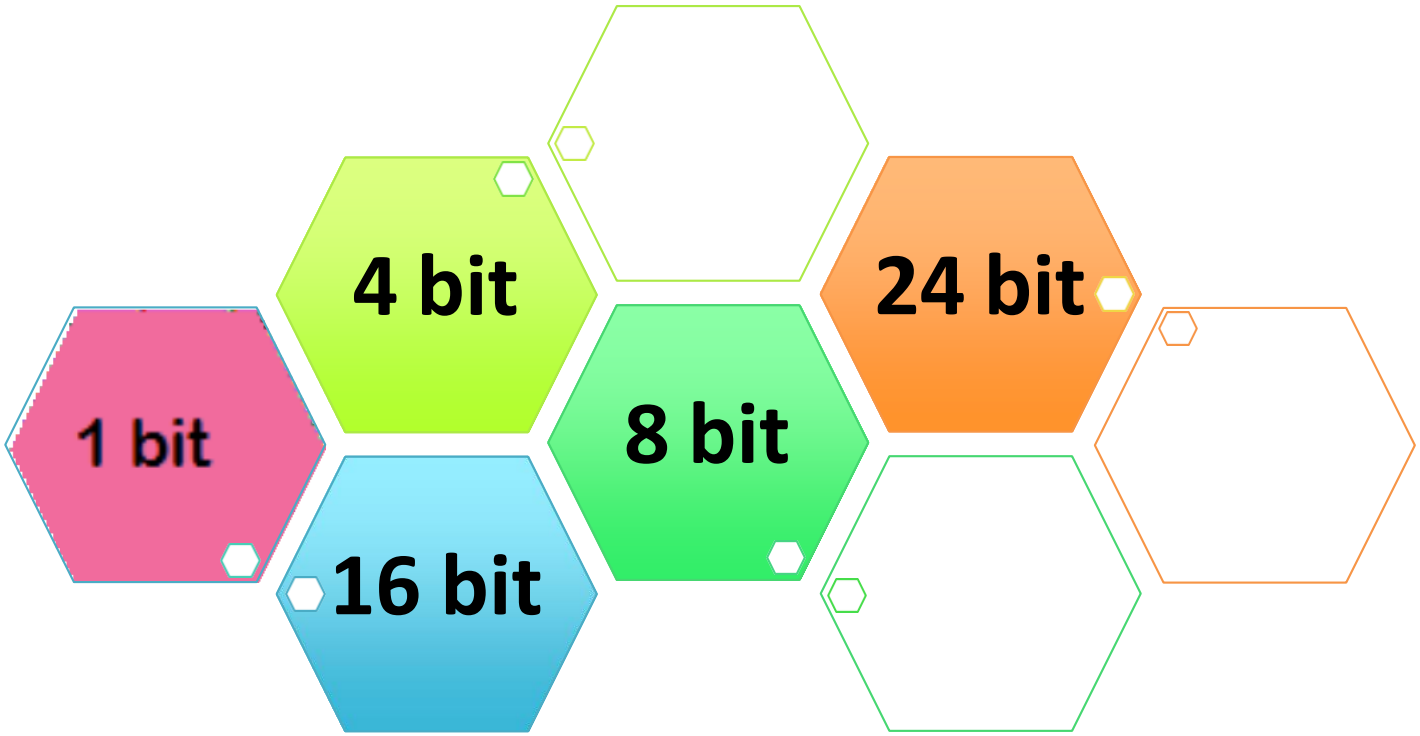
Merubah citra Gray ke Biner


```
import cv2

img=cv2.imread('parot.jpg',1)
cv2.imshow('Citra Burung',img)
# get dimensions of image
# height, width, number of channels in image
dimensionsRGB = img.shape
print('Image Dimension    : ',dimensionsRGB)

RGBtoGray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
cv2.imshow('Citra Burung Gray',RGBtoGray)
# height, width, number of channels in image
dimensionsGray = RGBtoGray.shape
print('Image Dimension Gray   : ',dimensionsGray)

# Gray to Binary
retval, threshold = cv2.threshold(RGBtoGray, 125, 255, cv2.THRESH_BINARY)
cv2.imshow('threshold',threshold)
dimensionsBinary = threshold.shape
print('Image Dimension Binary    : ',dimensionsBinary)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

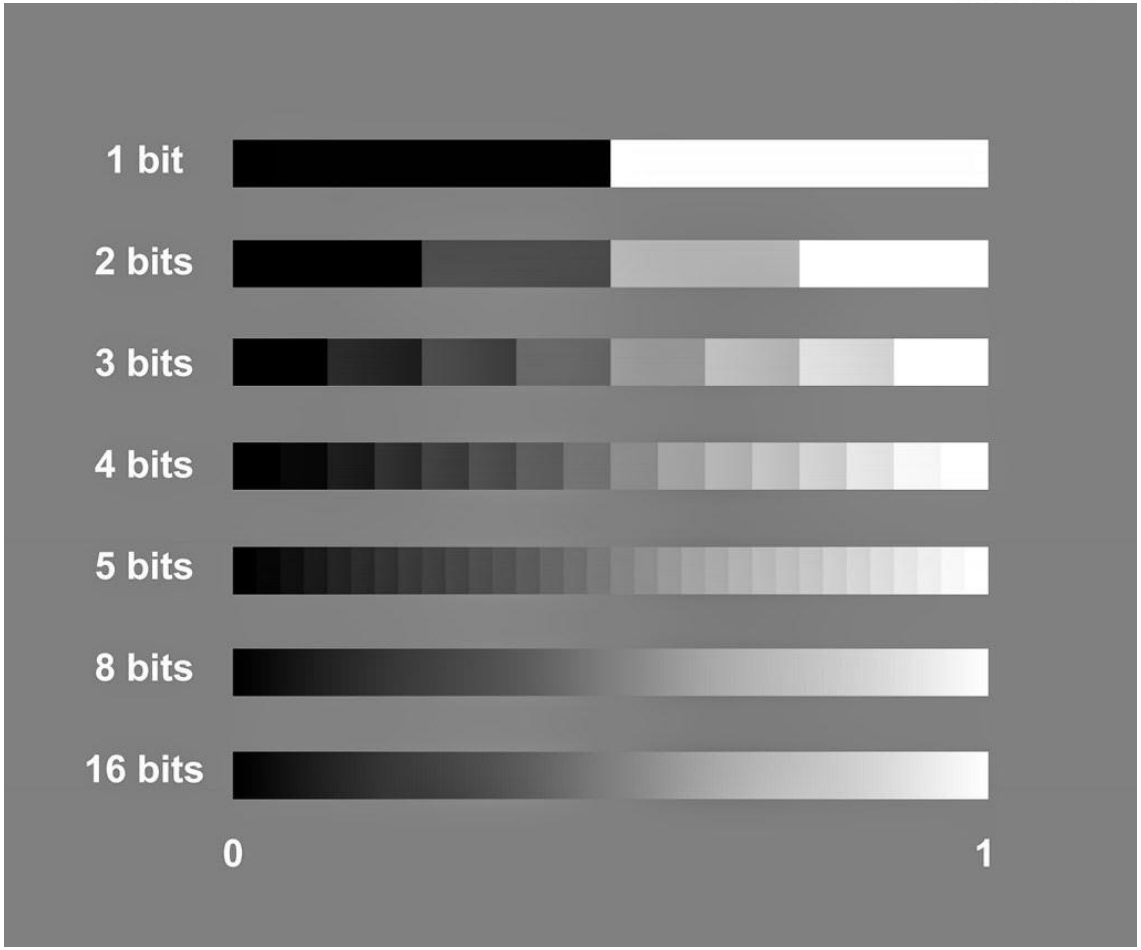
```
"C:\Users\Fitri Utaminingrum\Anaconda3\python.exe" "C:/Users/Fitri Utaminingrum/PycharmProjects/COBAUTS2/DimensiGray.py"
Image Dimension    :  (162, 308, 3)
Image Dimension Gray   :  (162, 308)
Image Dimension Binary    :  (162, 308)
```

Data Citra Digital

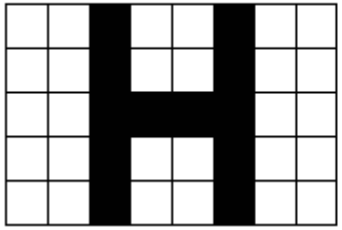
Sebuah citra digital dapat disusun oleh data citra digital mulai dari :



Citra Digital Hitam Putih



Contoh Data Citra Biner / 1 bit



Citra Biner (hitam = 0, putih = 1
= 1 1 0 1 1 0 1 1
= 1 1 0 1 1 0 1 1
= 1 1 0 0 0 0 1 1
= 1 1 0 1 1 0 1 1
= 1 1 0 1 1 0 1 1

Citra True Color

Citra Biner

Contoh Perbandingan Kualitas

Semakin lebar data citra digital maka kualitasnya semakin bagus

1-bit 4-bit 8-bit

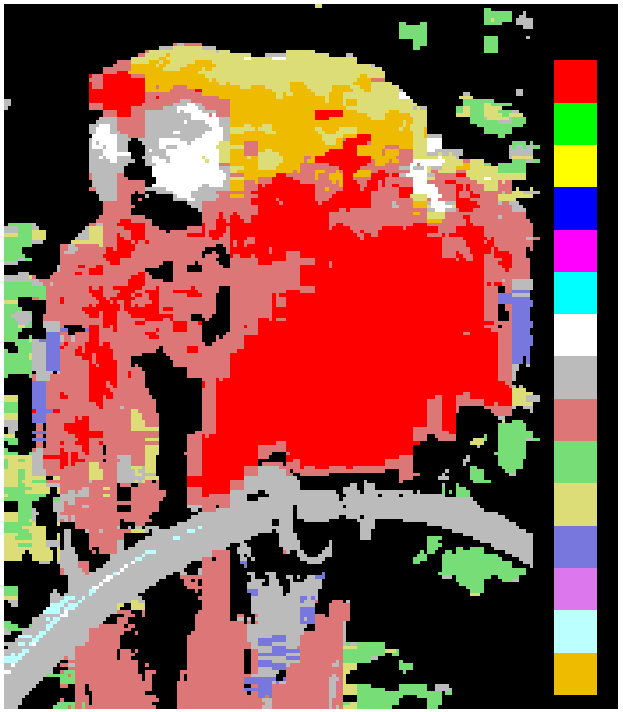
Data Citra Berwarna

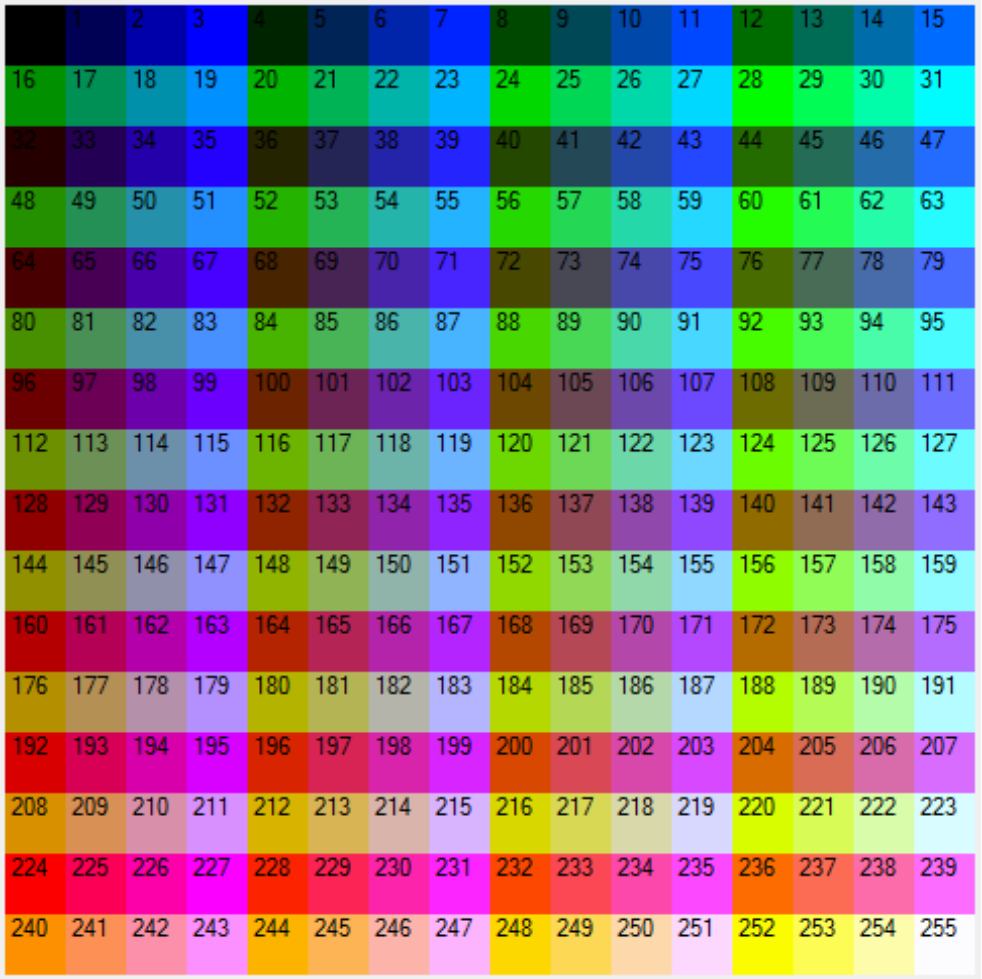
- 4 bit
- 8 bit
- 16 bit
- 24 bit

Kombinasi Berwarna 4 bit

Black 0, 0, 0	**Dark Red** 128, 0, 0	**Red** 255, 0, 0	**Pink** 255, 0, 255
Teal 0, 128, 128	**Green** 0, 128, 0	**Bright Green** 0, 255, 0	**Turquoise** 0, 255, 255
Dark Blue 0, 0, 128	**Violet** 128, 0, 128	**Blue** 0, 0, 255	**Gray 25%** 192, 192, 192
Gray 50% 128, 128, 128	**Dark Yellow** 128, 128, 0	**Yellow** 255, 255, 0	**White** 255, 255, 255

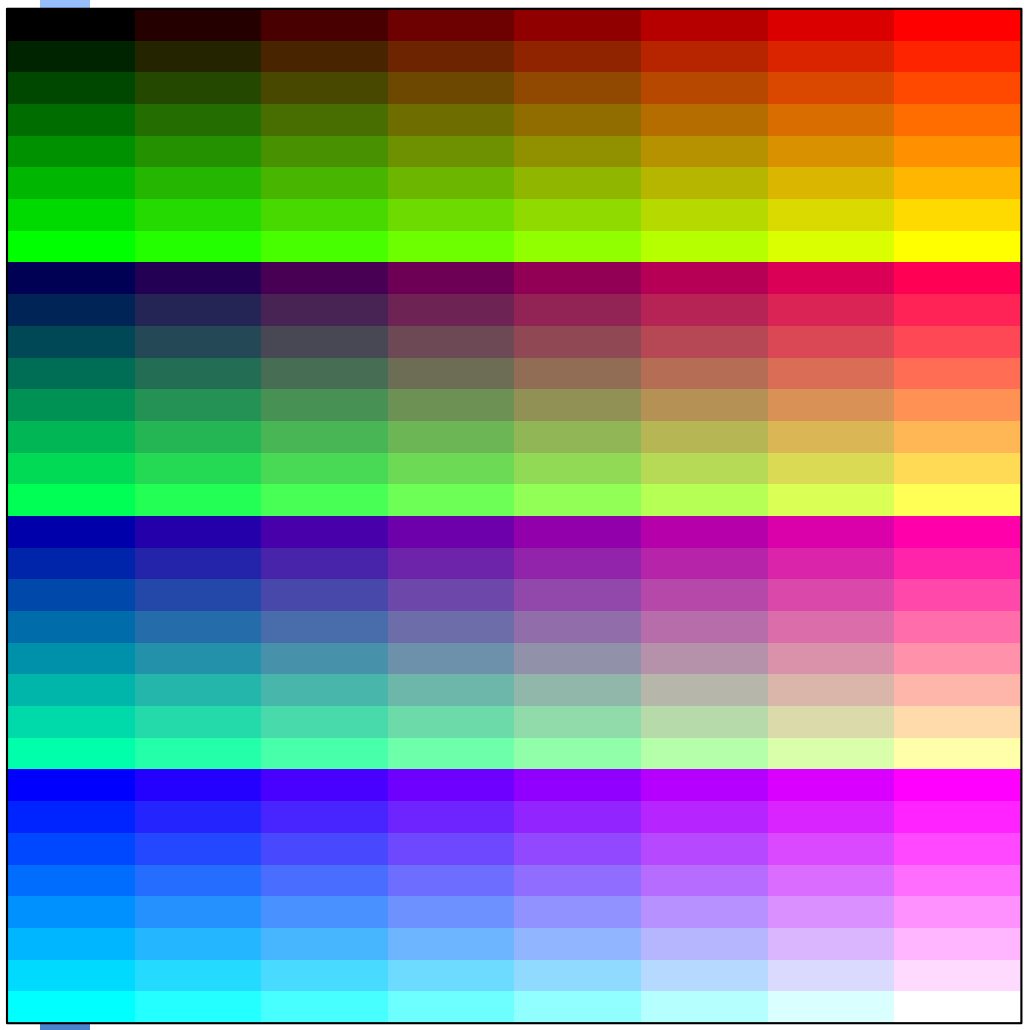
www.infotart.com





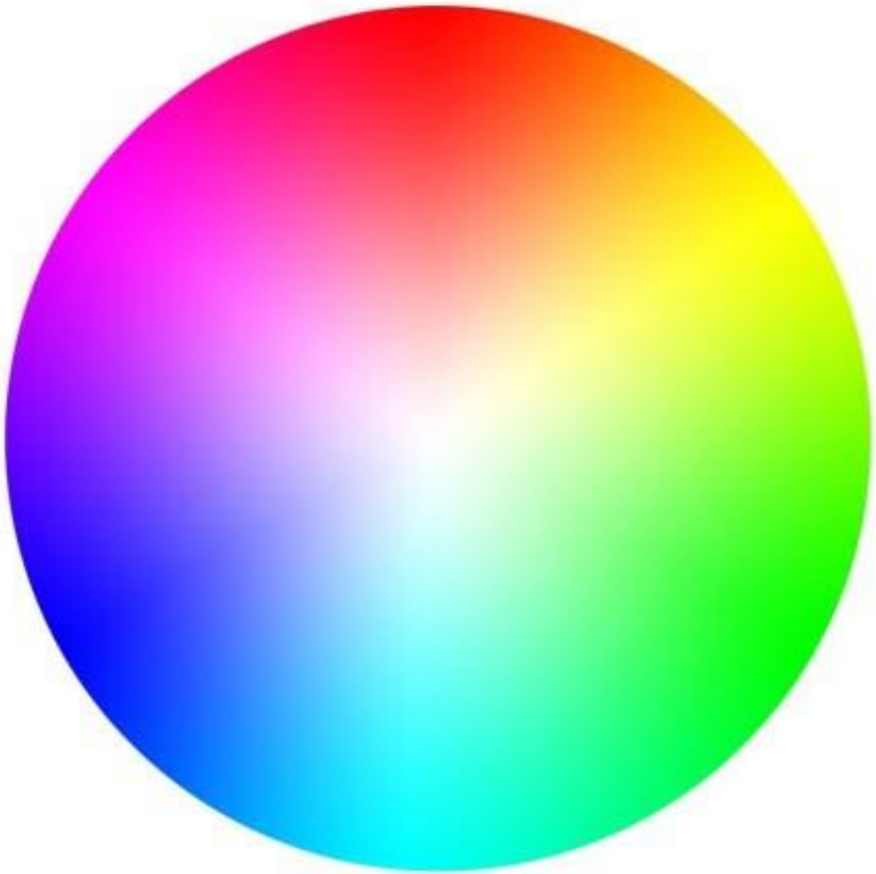
Kombinasi Berwarna 8 bit

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
R	R	R	G	G	G	B	B



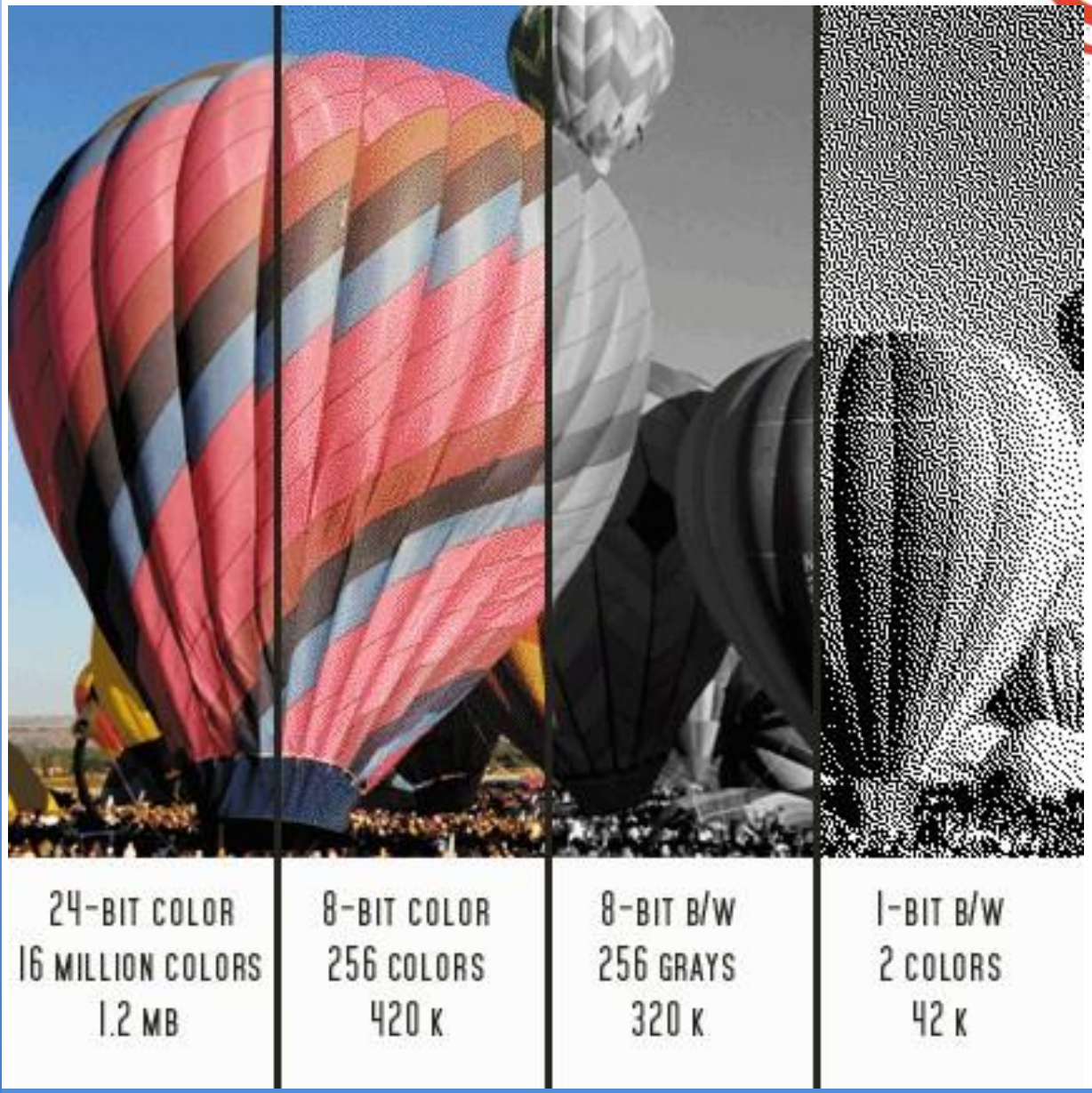
Kombinasi Berwarna 16 bit

Bit – 15	Bit – 14	Bit – 13	Bit – 12	Bit – 11	Bit – 10	Bit – 9	Bit – 8	Bit – 7	Bit – 6	Bit – 5	Bit – 4	Bit – 3	Bit – 2	Bit – 1	Bit – 0
R	R	R	R	R	G	G	G	G	G	G	B	B	B	B	B



Kombinasi Berwarna 24 bit

- Kombinasi Warna 24 bit disusun oleh 8-bit untuk R, 8-bit untuk G dan 8-bit untuk B



Perbandingan kualitas citra untuk lebar bit data yang berbeda

MODIFIKASI KECEMERLANGAN

- Pada dasarnya merubah nilai keabuan/warna dari gelap menuju terang adalah menggunakan formula linier :

Grey Scale : Po(x,y) = Pi(x,y) + C

Ket :

Po(x,y) = Pixel Output pada koordinat x,y

Pi(x,y) = Pixel Input pada koordinat x,y

C = Constanta

jika C positif maka lbh terang, C negatif lbh gelap

Untuk citra true color tentunya formula/rumus diubah dengan melibatkan RGB-nya :

Ro(x,y) = Ri(x,y) + C

Go(x,y) = Gi(x,y) + C

Bo(x,y) = Bi(x,y) + C

MODIFIKASI KECEMERLANGAN

Modifikasi Kecemerlangan f'(x,y) = f(x,y) + C ...dengan C = -100

NEGASI

- Merubah citra asli menjadi negasinya itu konsepnya adalah membalikan warna dengan batas 0 dan 255, artinya jika punya warna 0 menjadi 255 dan jika punya warna 255 menjadi 0 , formula liniernya sbb :

$$Po(x,y) = 255 - Pi(x,y)$$

Ket :

Po(x,y) = Pixel Output pada koordinat x,y

Pi(x,y) = Pixel Input pada koordinat x,y

NEGASI

Operasi Negasi

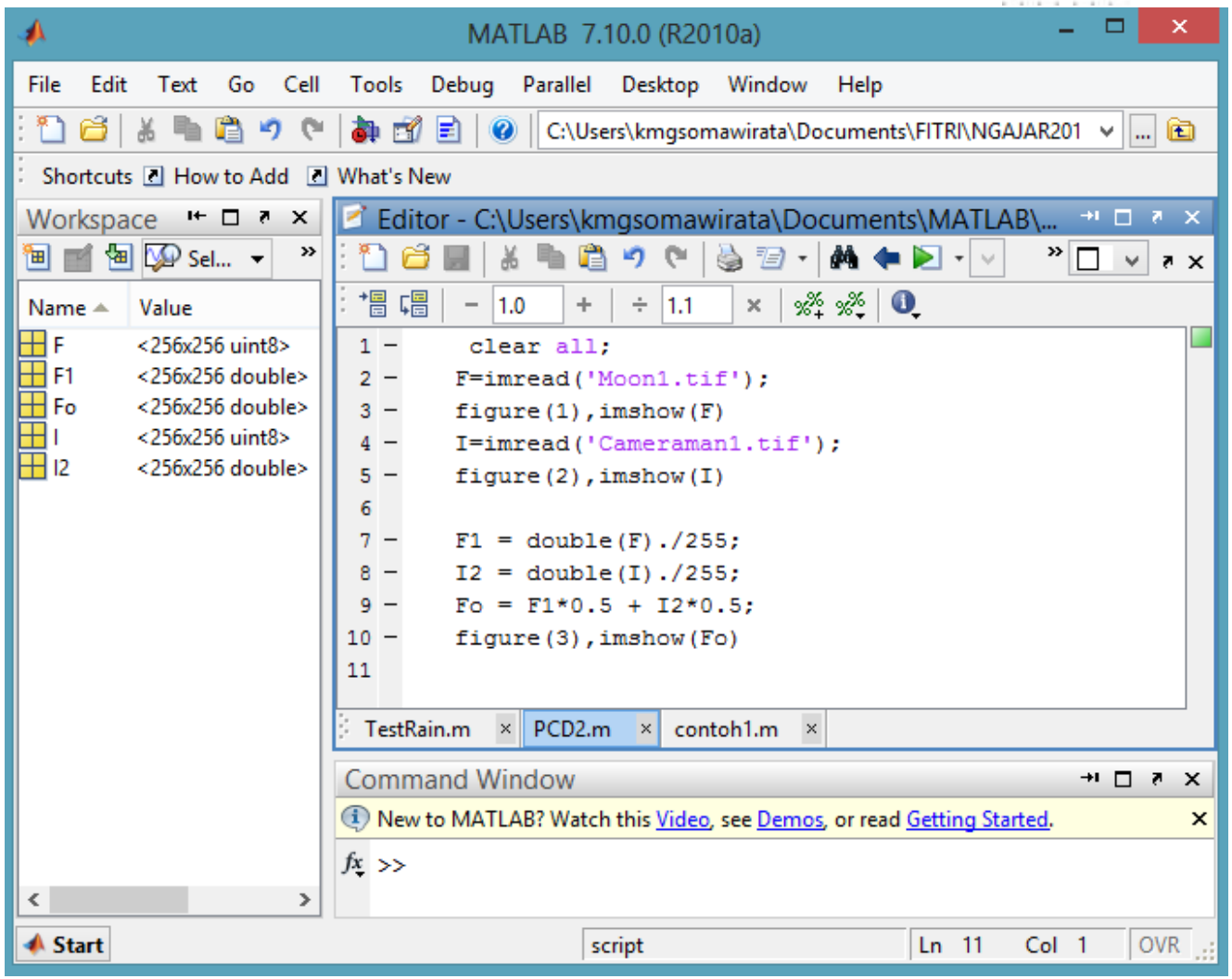
OPERASI DASAR PADA MATRIK CITRA

Operasi Penjumlahan

Syarat citra dapat dijumlahkan:

- Ukuran dan dimensi dari citra harus sama
- Nilai bobot (w1 dan w2) jika dijumlahkan sama dengan satu, Jika kurang dari satu maka citra hasil akan lebih gelap dan sebaliknya.
- Format data citra dari uint8 harus dirubah ke double

OPERASI PENJUMLAHAN DALAM MATLAB



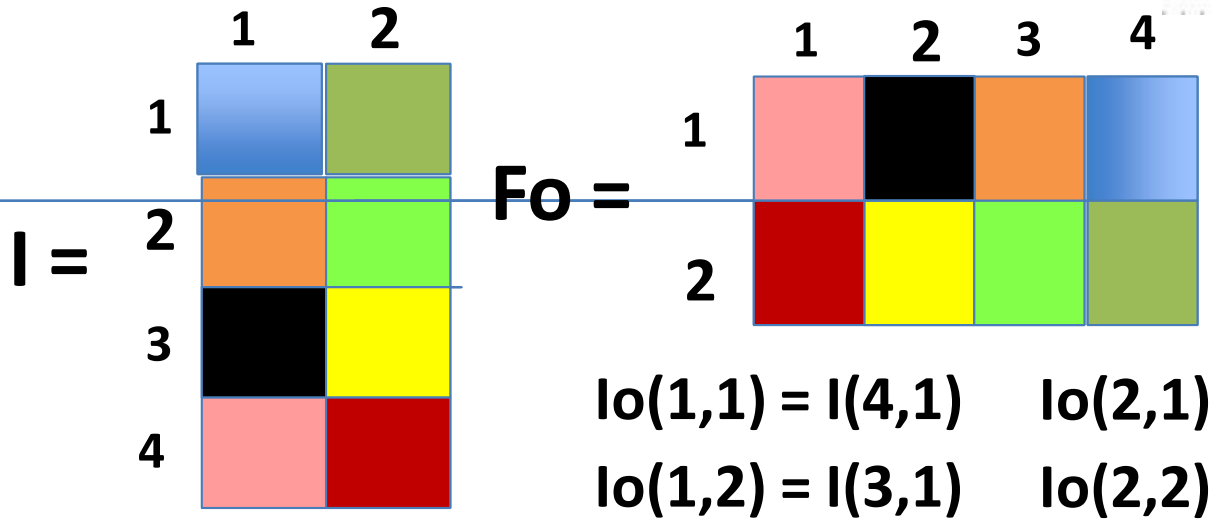
Operasi Penukaran baris menjadi kolom/Matrik Transpose

I

Fo

- Baris terakhir menjadi kolom pertama pada citra output
- Kolom pertama menjadi baris pertama pada citra output

ANALISA



Io(1,1) = I(4,1) **Io(2,1) = I(4,2)**

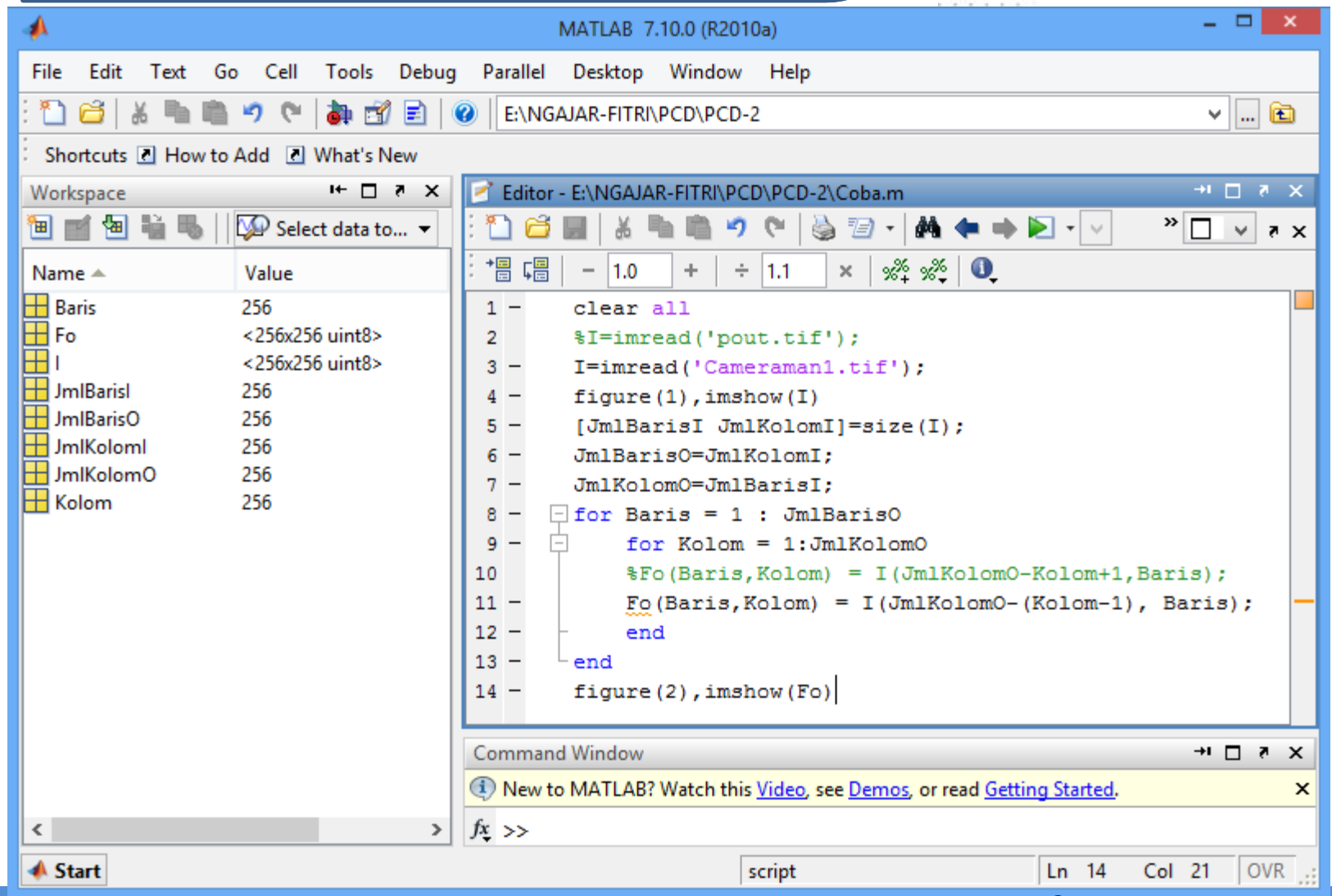
Io(1,2) = I(3,1) **Io(2,2) = I(3,2)**

Io(1,3) = I(2,1) **Io(2,3) = I(2,2)**

Io(1,4) = I(1,1) **Io(2,4) = I(1,2)**

- JmlBaris Fo=JlmKolom I dan JmlKolom Fo = JmlBaris I, sehingga dapat dirumuskan
  - Fo(Baris,Kolom) = I(JmlBarisI-(Kolom-1), Baris)
  - Fo(Baris,Kolom) = I(JmlBarisI-Kolom+1, Baris)

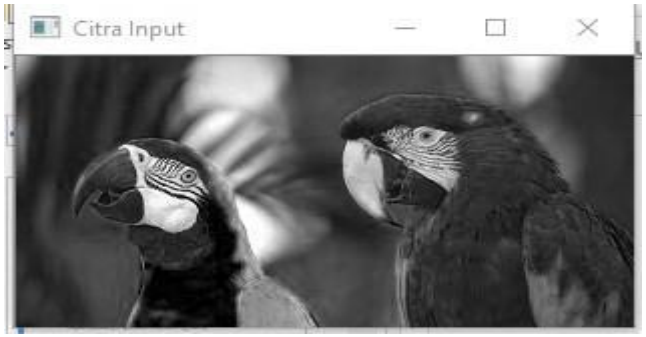
IMPLEMENTASI DALAM MATLAB



IMPLEMENTASI DALAM PYTHON

Rotate.py | RotasiManual.py | ContohFliping.py

```
3
4   img=cv2.imread('parot.jpg')
5   cv2.imshow('Citra Input',img[:,:,0])
6   Fo = np.ones((img.shape[1],img.shape[0]), dtype="uint8")
7
8   h1,w1 = img.shape[:2]
9
10  for i in range(Fo.shape[0]):
11      for j in range(Fo.shape[1]):
12          Fo[i,j]=img[h1-1-j,i,0]
13
14
15  cv2.imshow('Citra Hasil',Fo)
16  cv2.waitKey(0)
17  cv2.destroyAllWindows()
18
```



OPERASI PENCERMINAN

I

Fo

- Urutan baris untuk citra Fo = baris citra I
- Kolom pertama citra Fo = Kolom terakhir citra I

OPERASI PENCERMINAN ANALISA

- Jika citra masukkan I dan citra keluaran Fo

  - Fo(1,1) = I(1,256)
  - Fo(1,2) = I(1,255)
  - Fo(1,3) = I(1,254)
  - Fo(1,5) = I(1,253)
  - Fo(1,5) = I(1,252)
    - :
    - :
  - Fo(1,255) = I(1,2)
  - Fo(1,256) = I(1,1)

  - Fo(2,1) = I(2,256)
  - Fo(2,2) = I(2,255)
  - Fo(2,3) = I(2,254)
  - Fo(2,5) = I(2,253)
  - Fo(2,5) = I(2,252)
  - :
  - :
  - Fo(2,255) = I(2,2)
  - Fo(2,256) = I(2,1)

  - Fo(3,1) = I(3,256)
  - Fo(3,2) = I(3,255)
  - Fo(3,3) = I(3,254)
  - Fo(3,5) = I(3,253)
  - Fo(3,5) = I(3,252)
  - :
  - :
  - Fo(3,255) = I(3,2)
  - Fo(3,256) = I(3,1)

- Dan seterusnya

- Ukuran citra I= 256X256 sehingga JmlBaris=256 dan JmlKolom=256,
- JmlBaris Fo=JlmBaris I dan JmlKolom Fo = JmlKolom I, sehingga dapat dirumuskan
  - Fo(Baris,Kolom) = I(Baris,JmlKolom+1-Kolom)

IMPLEMENTASI PENCERMINAN

Editor - C:\Users\kmgsomawirata\Documents\FITRI\NGAJAR2016\...

```
1   function Fo= Pencerminan(fi)
2   [JmlBaris JmlKolom] = size(fi);
3   for Baris = 1 : JmlBaris
4       for Kolom = 1: JmlKolom
5           Fo(Baris,Kolom) = fi(Baris,JmlKolom+1-Kolom);
6
7       end
8   end
9   return
10
```

IMPLEMENTASI PENCERMINAN PYTHON

Rotate.py | RotasiManual.py | PencerminanManual.py | Co

```
1   import cv2
2   import numpy as np
3
4   img=cv2.imread('parot.jpg')
5   cv2.imshow('Citra Input',img[:,:,0])
6   Fo = np.ones((img.shape[0],img.shape[1]), dtype="uint8")
7
8   hl,wl = img.shape[:2]
9
10  for i in range(img.shape[0]):
11      for j in range(img.shape[1]):
12          Fo[i,j]=img[i,wl-1-j,0]
13
14
15  cv2.imshow('Citra Hasil',Fo)
16  cv2.waitKey(0)
17  cv2.destroyAllWindows()
18
```

CROPPING

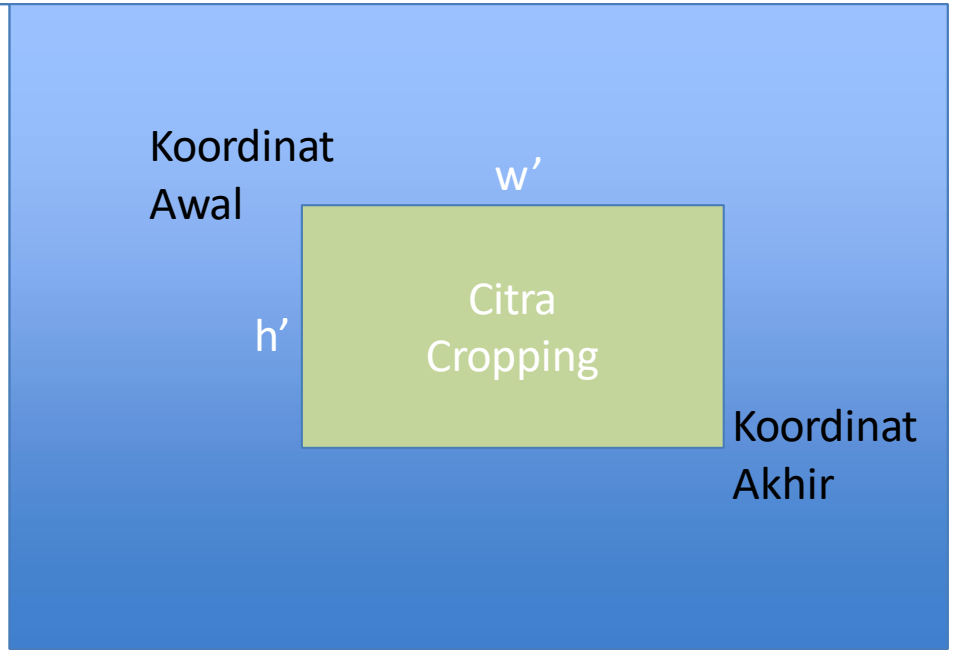
Cropping adalah memotong sebagian citra menjadi ukuran yang lebih kecil

KoordinatAwal(x1,y1)

KoordinatAkhir(x2,y2)

h’=x2-x1

w’=y2-y1



CROPPING

Rotate.py | RotasiManual.py | PencerminanManual.py | Cropping.py

```
1   import cv2
2   import numpy as np
3
4   img=cv2.imread('parot.jpg')
5   cv2.imshow('Citra Input',img[:,:,0])
6
7   koAwal = [50,50]
8   koAkhir = [150,200]
9   Fo = np.ones((koAkhir[0]-koAwal[0],koAkhir[1]-koAwal[1]), dtype="uint8")
10
11  h1,w1 = img.shape[:2]
12
13  for i in range(Fo.shape[0]):
14      for j in range(Fo.shape[1]):
15          Fo[i,j]=img[i+koAwal[0],j+koAwal[1],0]
16
17
18  cv2.imshow('Citra Hasil',Fo)
19  cv2.waitKey(0)
20  cv2.destroyAllWindows()
21
```