

PERTEMUAN III PRAKTIKUM CITRA

Format Citra, Layer RGB, dan Gray Scale

Nana Ramadijanti



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
2021

Praktikum III

Format Citra, Layer RGB, dan Gray Scale

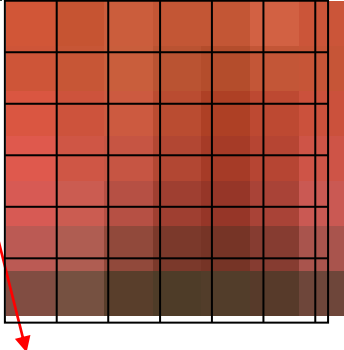


MATERI

1. Membaca data citra RGB
2. Representasi Obyek Citra (Matrik 2D dari Intensitas Piksel)
3. Konversi data RGB
4. Obyek Citra dari Numpy
5. Percobaan Layer BGR, Grayscale dan Biner



Citra RGB



Citra Digital = sebuah array multidimensi
(misal berisi intensitas citra)

Setiap komponen pada citra
disebut piksel dinyatakan
Dengan nilai piksel

10	10	16	28
9	65	70	56
15	32	99	70
32	21	60	90
	54	85	85
		32	65



Citra RGB (2)



Citra Berwarna atau RGB image:

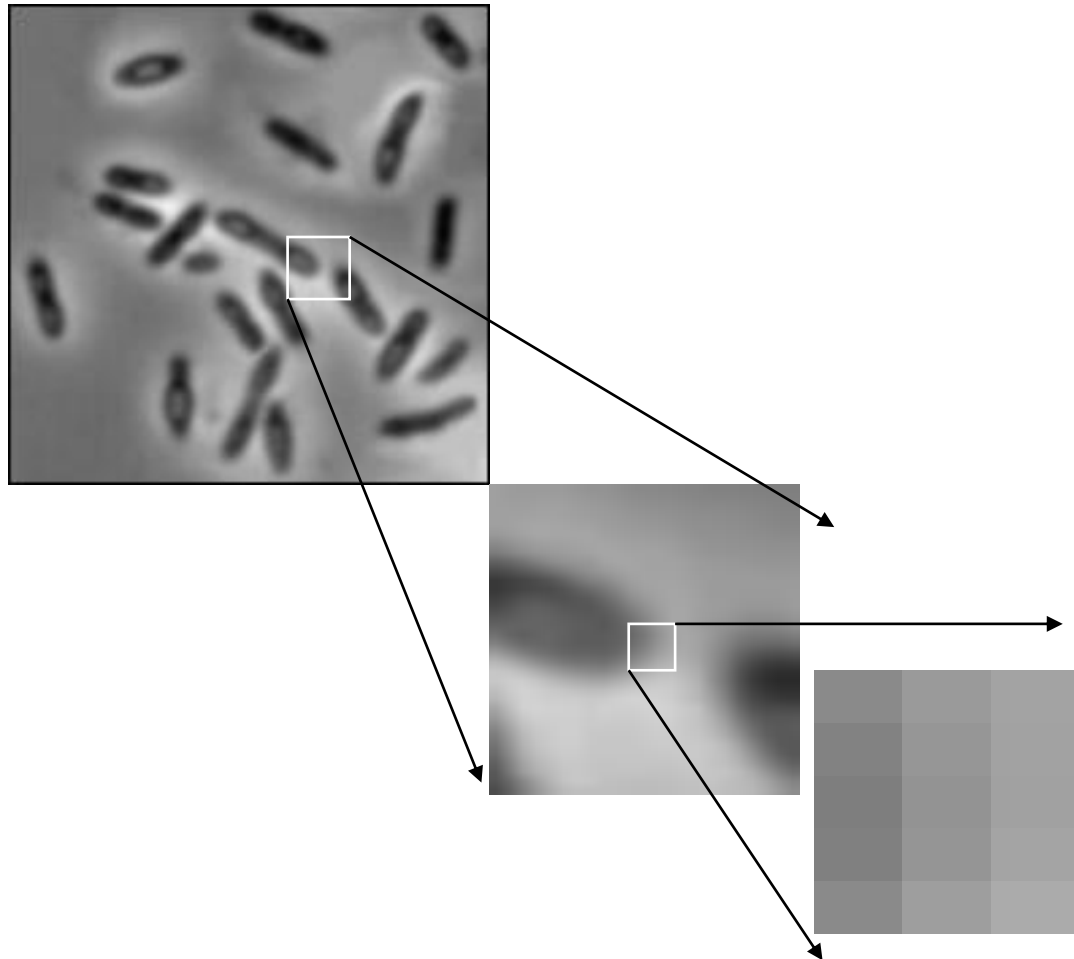
Setiap piksel berisi sebuah vektor yang merepresentasikan komponen red, green, dan blue

Komponen
RGB

10	10	16	28
9	65	70	56
15	32	99	70
32	21	60	90
	54	85	85
		32	65



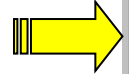
Citra Gray Scale



Intensitas Citra atau Citra Monochrome/grayscale

Setiap piksel berhubungan dengan intensitas yang biasa disebut grayscale (gray level/keabuan)

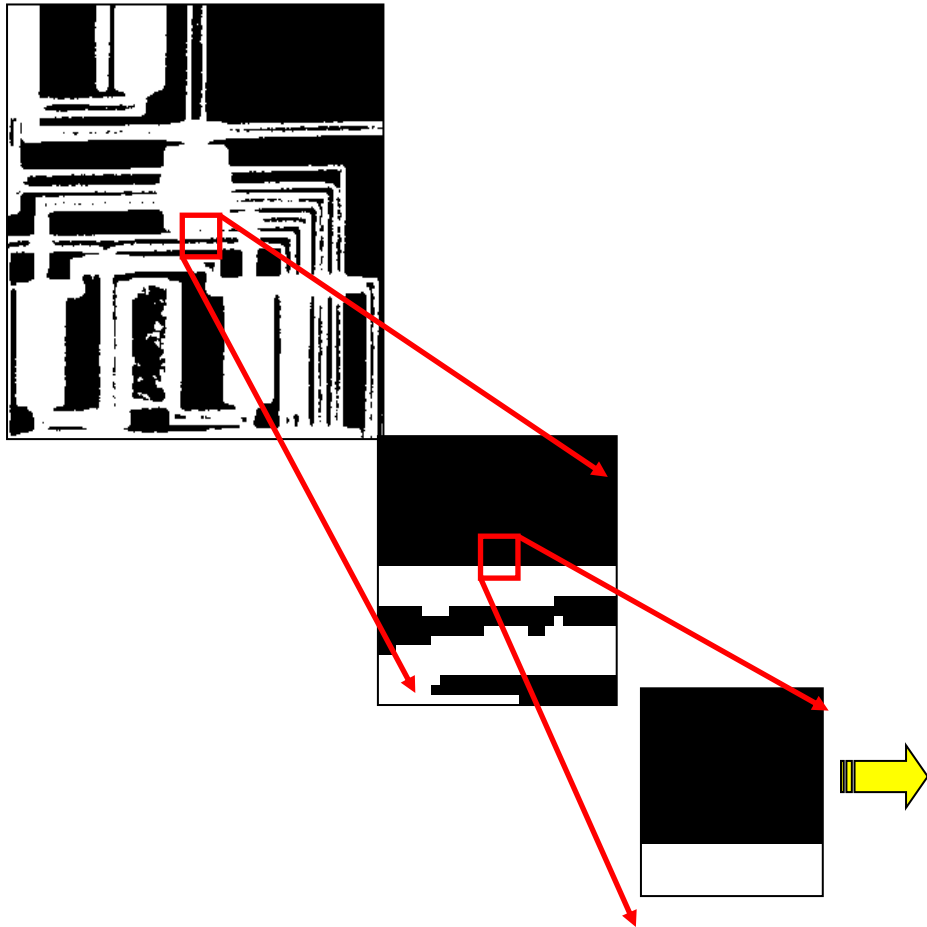
Nilai Gray Scale



10	10	16	28
9	6	26	37
15	25	13	22
32	15	87	39



Citra Biner



Citra Biner atau Citra Hitam Putih :

Masing-masing piksel berisi 1 piksel :

1 Merepresentasikan putih

0 Merepresentasikan hitam

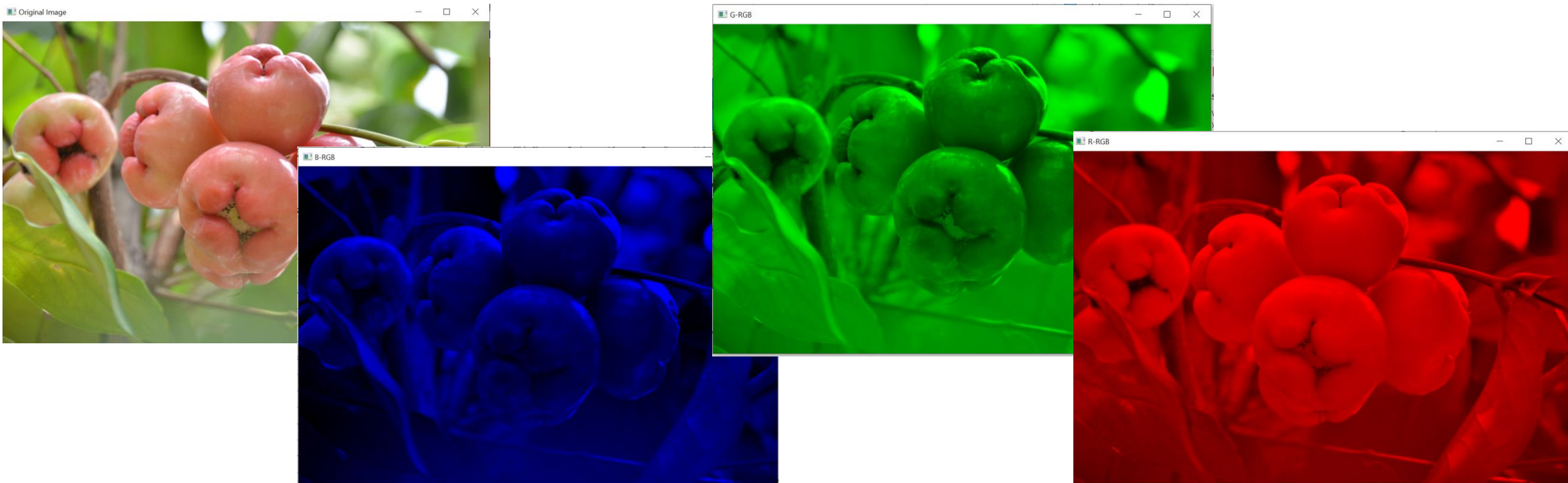
Data Biner

0	0	0	0
0	0	0	0
1	1	1	1
1	1	1	1



Citra sebagai Matrik 2D Layer B/G/R

Percobaan ini digunakan untuk mengambil nilai dari layer B, dan kemudian ditampilkan dengan meng-nol-kan layer G dan layer R. Dilanjutkan menampilkan layer G, dengan memberikan nilai 0 pada matrik layer B dan memberikan nilai 0 pada matrik layer B dan matrik layer G,





Percobaan 1 : Layer B/G/R

```
import cv2
# Membaca data Image
img = cv2.imread("jambu.jpg")

b=img.copy()
b[:, :, 1]=0
b[:, :, 2]=0

g=img.copy()
g[:, :, 0]=0
g[:, :, 2]=0
cv2.imshow('G-RGB',g)

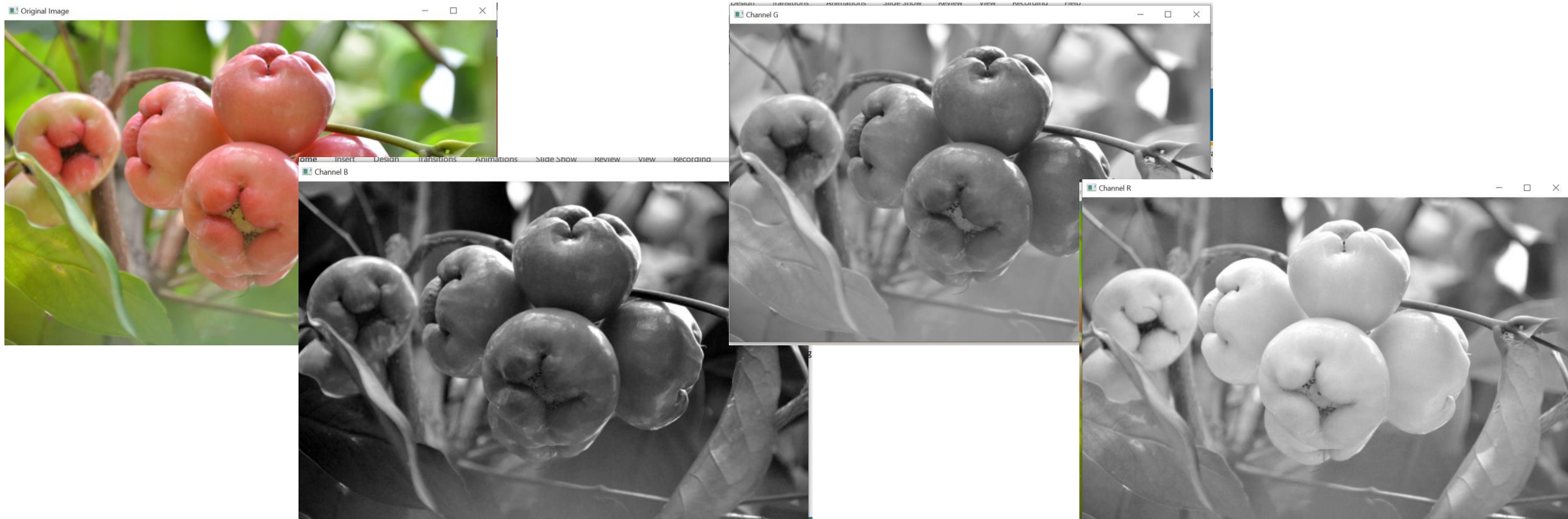
r=img.copy()
r[:, :, 0]=0
r[:, :, 1]=0

cv2.imshow('Original Image',img)
cv2.imshow('B-RGB',b)
cv2.imshow('G-RGB',g)
cv2.imshow('R-RGB',r)
```




Citra sebagai Matrik 2D Grayscale

Pada dasarnya citra grayscale atau derajat keabuan adalah citra yang menggunakan nilai B, G dan R yang sama, misalkan (100,100,100) atau (140,140,140) atau (xg, xg, xg) . Dengan demikian bila diambil salah satu warna b, g atau r, dan dijadikan warna baru dengan (r, r, r) atau (g, g, g) atau (b, b, b) , akan juga menghasilkan gambar grayscale.





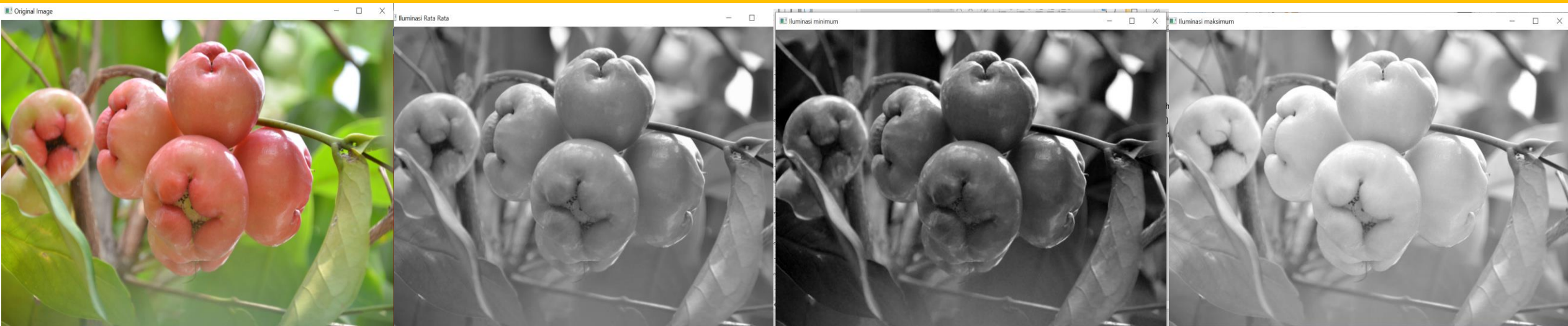
Percobaan 2 : Grayscale dari B/G/R

```
import cv2
# Membaca data Image
img = cv2.imread("jambu.jpg")

B , G , R = cv2.split(img)
cv2.imshow("Original Image", img)
cv2.imshow("Channel R", R)
cv2.imshow("Channel G", G)
cv2.imshow("Channel B", B)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



Citra Grayscale dengan Iluminasi yang Berbeda



Citra RGB

Mempunyai komponen r, g dan b yang masing-masing bernilai 0 s/d 255

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

Citra Grayscale :

menggunakan nilai rata-rata dari r, g dan b

Citra Grayscale :

menggunakan nilai minimum dari r, g dan b

$$x = \min(r, g, b)$$

Citra Grayscale :

menggunakan nilai maksimum dari r, g dan b

$$x = \max(r, g, b)$$



Percobaan 3 :

Grayscale dengan Iluminasi Citra

```
import cv2
import numpy as np
img = cv2.imread("jambu.jpg")
B , G , R = cv2.split(img)

img_gray1 = 0.33 * R + 0.33 * G + 0.33 * B
img_gray1 = img_gray1.astype(np.uint8)
img_RG1 = np.minimum(R, G)
img_gray2 = np.minimum(img_RG1 , B)
img_RG2 = np.maximum(R , G)
img_gray3 = np.maximum(img_RG2 , B)
cv2.imshow("Iluminasi Rata Rata", img_gray1)
cv2.imshow("Iluminasi minimum", img_gray2)
cv2.imshow("Iluminasi maksimum", img_gray3)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



Citra Binary dengan Parameter Threshold

Binary

$$\text{dst}(x, y) = \begin{cases} \text{maxval} & \text{if } \text{src}(x, y) > \text{thresh} \\ 0 & \text{otherwise} \end{cases}$$

Inverted Binary

$$\text{dst}(x, y) = \begin{cases} 0 & \text{if } \text{src}(x, y) > \text{thresh} \\ \text{maxval} & \text{otherwise} \end{cases}$$

Truncated

$$\text{dst}(x, y) = \begin{cases} \text{threshold} & \text{if } \text{src}(x, y) > \text{thresh} \\ \text{src}(x, y) & \text{otherwise} \end{cases}$$

To Zero

$$\text{dst}(x, y) = \begin{cases} \text{src}(x, y) & \text{if } \text{src}(x, y) > \text{thresh} \\ 0 & \text{otherwise} \end{cases}$$

To Zero Inverted

$$\text{dst}(x, y) = \begin{cases} 0 & \text{if } \text{src}(x, y) > \text{thresh} \\ \text{src}(x, y) & \text{otherwise} \end{cases}$$

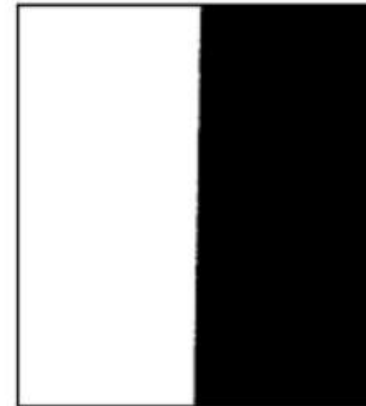
Original Image



BINARY



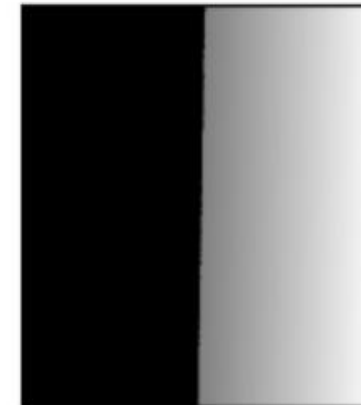
BINARY_INV



TRUNC



TOZERO



TOZERO_INV





Percobaan 4 : Citra Binary

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
from urllib3.connectionpool import xrange

img = cv2.imread('gradient.png',0)
ret,thresh1 = cv2.threshold(img,127,255,cv2.THRESH_BINARY)
ret,thresh2 = cv2.threshold(img,127,255,cv2.THRESH_BINARY_INV)
ret,thresh3 = cv2.threshold(img,127,255,cv2.THRESH_TRUNC)
ret,thresh4 = cv2.threshold(img,127,255,cv2.THRESH_TOZERO)
ret,thresh5 = cv2.threshold(img,127,255,cv2.THRESH_TOZERO_INV)

titles = ['Original Image','BINARY','BINARY_INV','TRUNC','TOZERO','TOZERO_INV']
images = [img, thresh1, thresh2, thresh3, thresh4, thresh5]

for i in xrange(6):
    plt.subplot(2,3,i+1),plt.imshow(images[i],'gray')
    plt.title(titles[i])
    plt.xticks([],plt.yticks([]))
plt.show()
cv2.waitKey(0)
cv2.destroyAllWindows()
```




Praktikum III

Format Citra, Layer RGB, dan Gray Scale

TUGAS LAPORAN PRAKTIKUM

- (1) Tambahkan sebuah program untuk proses “Sephia” dengan mengambil nilai r dari layer R, kemudian pada warna yang baru ganti nilai R dengan $2*r$, nilai G dengan $1.8*r$ dan nilai B dengan r .
- (2) Tuliskan semua kode program percobaan 1-4 dari praktikum tentang manipulasi RGB
- (3) Lakukan percobaan layer BGR, grayscale, dan binar dengan gambar-gambar berikut, dan laporkan hasilnya serta berikan penjelasan mengapa hasilnya seperti yang tercetak di laporan.

Gambar RGB untuk Percobaan





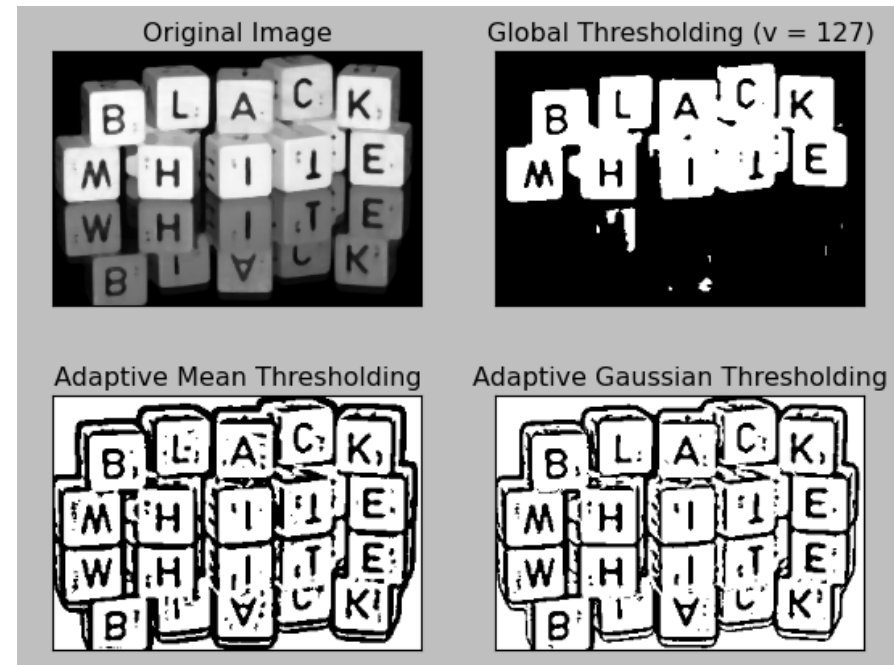
Praktikum III

Format Citra, Layer RGB, dan Gray Scale

TUGAS LAPORAN PRAKTIKUM

(4) Menggunakan nilai threshold global mungkin bukan pilihan yang baik di mana gambar memiliki kondisi pencahayaan yang berbeda di area yang berbeda. Buatlah program thresholding adaptif. Menggunakan algoritma yang menghitung batas threshold untuk wilayah kecil gambar sehingga kita bisa mendapatkan batas threshold yang berbeda untuk wilayah yang berbeda dari gambar yang sama dan memberi kita hasil yang lebih baik untuk gambar dengan kondisi cahaya yang bervariasi.

Output Adaptive Thresholding



```
cv.AdaptiveThreshold(src, dst, maxValue,  
adaptive_method=CV_ADAPTIVE_THRESH_MEAN_C,  
thresholdType=CV_THRESH_BINARY, blockSize=3, param1=5)
```


Praktikum III

Format Citra, Layer RGB, dan Gray Scale



LAPORAN PRAKTIKUM

1. Tuliskan koding lengkap dari program percobaan dan tugas laporan di atas
2. Jelaskan fungsi dan parameter di CV2 yang berfungsi untuk thresholding bindary dan adaptive thresholding:

```
cv2.threshold(src, thresh, maxval, type[, dst])
```

```
cv.AdaptiveThreshold(src, dst, maxValue, adaptive_method=CV_ADAPTIVE_THRESH_MEAN_C,  
thresholdType=CV_THRESH_BINARY, blockSize=3, param1=5)
```

3. Berikan analisa dan keterangan pada setiap baris program yang anda anggap penting.

Terima Kasih



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA