

Nama : Muhammmad Zidan Fadillah

Nim : 312210277

Kelas : TI.22.A.2

Mata Kuliah : Pengolahan Citra

---

## Manipulasi Citra Digital

- **Modul**

import cv2

- **Original Image**

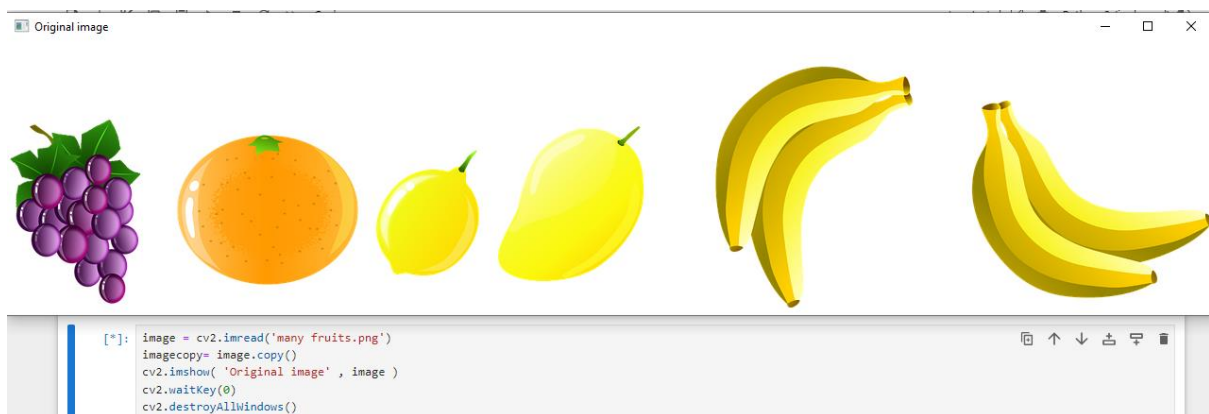
```
image = cv2.imread('many fruits.png')
```

```
imagecopy= image.copy()
```

```
cv2.imshow( 'Original image' , image )
```

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

```
cv2.destroyAllWindows()
```



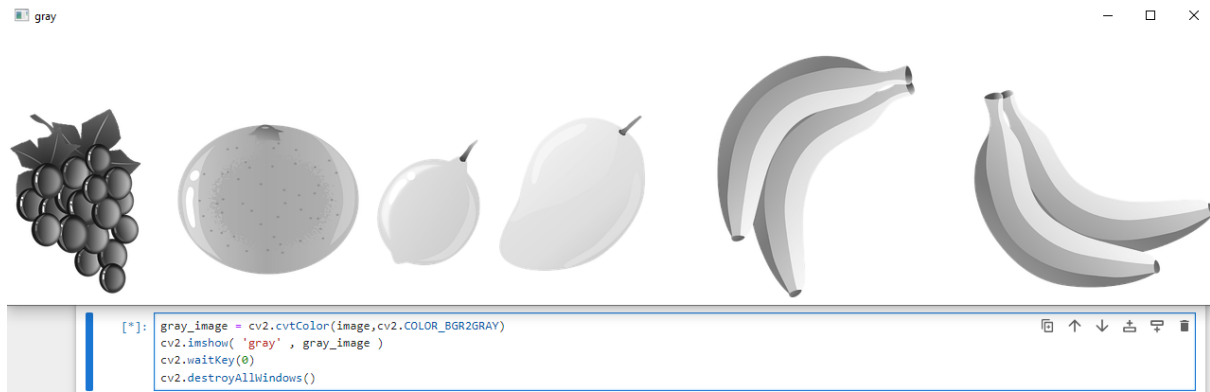
- **gray**

```
gray_image = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
```

```
cv2.imshow( 'gray' , gray_image )
```

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

```
cv2.destroyAllWindows()
```



- **binary**

```
ret,binary_im = cv2.threshold(gray_image,245,255,cv2.THRESH_BINARY)
```

```
cv2.imshow( 'binary' , binary_im )
```

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

```
cv2.destroyAllWindows()
```



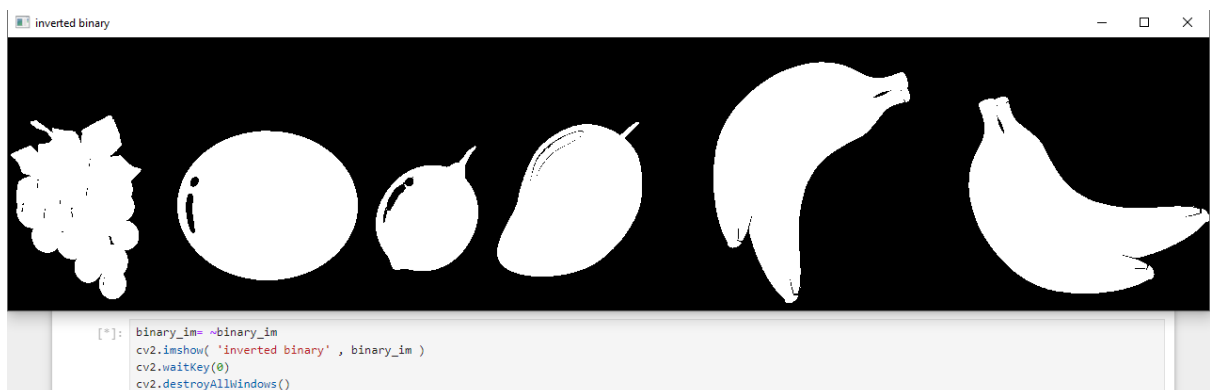
- **inverted binary**

```
binary_im= ~binary_im
```

```
cv2.imshow( 'inverted binary' , binary_im )
```

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

```
cv2.destroyAllWindows()
```



```
contours,hierarchy  
cv2.findContours(binary_im,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)
```

- **contours marked on RGB image**

```
with_contours = cv2.drawContours(image,contours,-1,(0,0,255),3)  
cv2.imshow( 'contours marked on RGB image' , with_contours )  
cv2.waitKey(0)  
cv2.destroyAllWindows()
```



- **Reference image**

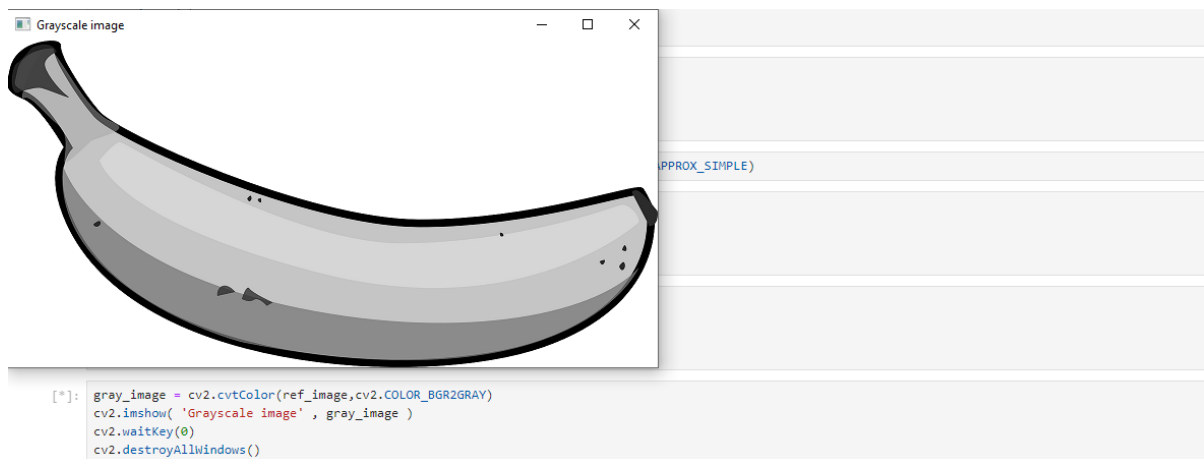
```
ref_image = cv2.imread('bananaref.png')  
cv2.imshow( 'Reference image' , ref_image )  
cv2.waitKey(0)  
cv2.destroyAllWindows()
```



- **Grayscale image**

```
gray_image = cv2.cvtColor(ref_image,cv2.COLOR_BGR2GRAY)  
cv2.imshow( 'Grayscale image' , gray_image )  
cv2.waitKey(0)
```

cv2.destroyAllWindows()



- **Binary image**

```
ret,binary_im = cv2.threshold(gray_image,245,255,cv2.THRESH_BINARY)
```

```
cv2.imshow( 'Binary image' , binary_im )
```

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

```
cv2.destroyAllWindows()
```



- **inverted binary image**

```
binary_im= binary_im
```

```
cv2.imshow( 'inverted binary image' , binary_im )
```

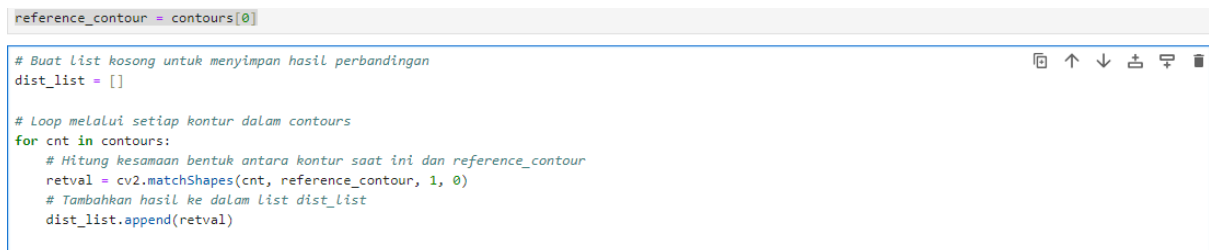
```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```



•

reference\_contour = contours[0]



### • Contours marked on RGB image

reference\_contour = contours[0]

# Buat list kosong untuk menyimpan hasil perbandingan

dist\_list = []

# Loop melalui setiap kontur dalam contours

for cnt in contours:

    # Hitung kesamaan bentuk antara kontur saat ini dan reference\_contour

    retval = cv2.matchShapes(cnt, reference\_contour, 1, 0)

    # Tambahkan hasil ke dalam list dist\_list

    dist\_list.append(retval)

sorted\_list= dist\_list.copy()

sorted\_list.sort() # sorts the list from smallest to largest

ind1\_dist= dist\_list.index(sorted\_list[0])

ind2\_dist= dist\_list.index(sorted\_list[1])

banana\_cnts= []

banana\_cnts.append(contours[ind1\_dist])

```

banana_cnts.append(contours[ind2_dist])

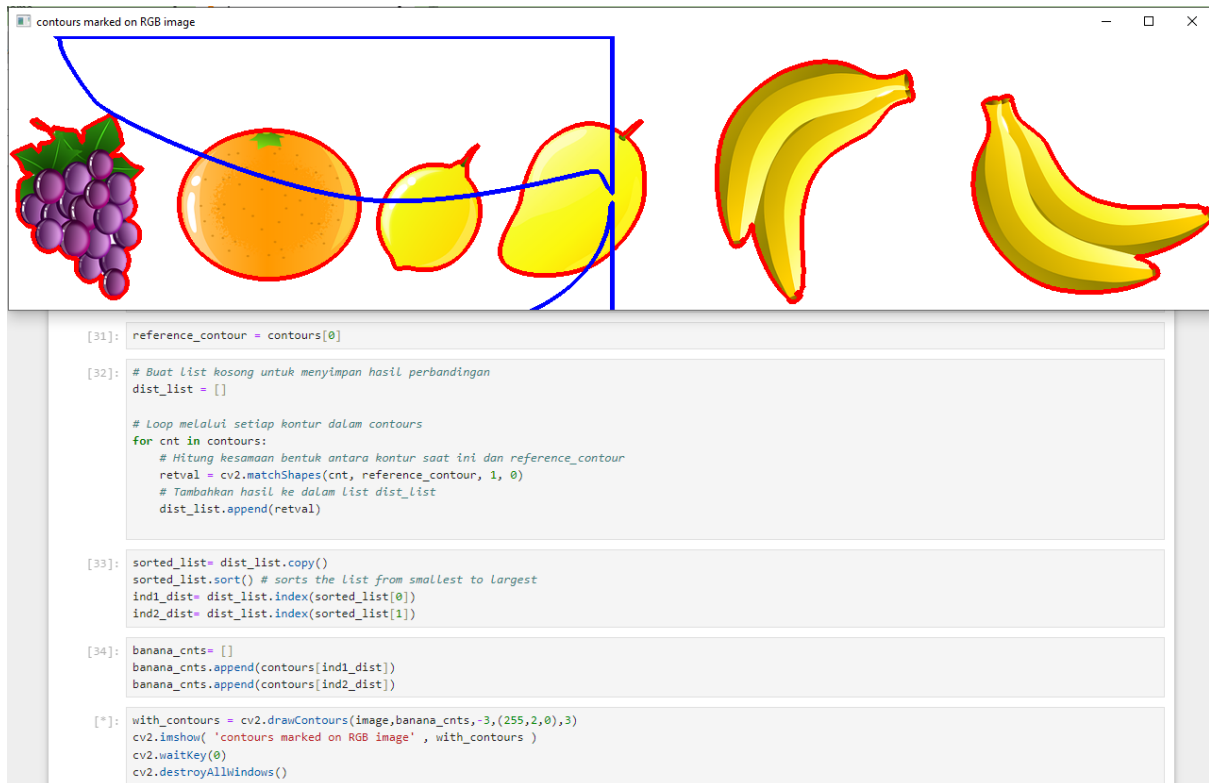
with_contours = cv2.drawContours(image,banana_cnts,-3,(255,2,0),3)

cv2.imshow( 'contours marked on RGB image' , with_contours )

cv2.waitKey(0)

cv2.destroyAllWindows()

```



- **Upright banana marked on RGB image**

```
for cnt in banana_cnts:
```

```
    x, y, w, h = cv2.boundingRect(cnt)
```

```
    if h > w:
```

```
        # Calculate the center and radius of the circle
```

```
        center = (x + w // 2, y + h // 2)
```

```
        radius = max(w, h) // 2
```

```
        # Draw the circle
```

```
        cv2.circle(imagecopy, center, radius, (255, 0, 0), 5)
```

```
cv2.imshow('Upright banana marked on RGB image', imagecopy)
```

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

cv2.destroyAllWindows()

