

PERTEMUAN 2. PERBAIKAN CITRA PADA PIXEL

BAGIAN 1. Grayscale, Histogram (Latihan2a_NIM.ipynb)

1. Load the Library

Bagian ini akan memuat library yang digunakan di dalam notebook: numpy, pandas, cv2, skimage, PIL, matplotlib

* Numpy (<https://www.numpy.org/>) adalah pustaka manipulasi array, digunakan untuk aljabar linier, transformasi Fourier, dan kemampuan bilangan acak.

* Pandas (<https://pandas.pydata.org/>) adalah library untuk manipulasi data dan analisis data.

* CV2 (https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_gui/py_image_display/py_image_display.html) adalah library untuk tugas-tugas computer vision.

* Skimage (<https://scikit-image.org/>) adalah library yang mendukung aplikasi pengolah gambar pada python.

* Matplotlib (<https://matplotlib.org/>) adalah library yang menghasilkan gambar dan menyediakan toolkit antarmuka pengguna grafis.

```
1 import numpy as np
2 import pandas as pd
3 import cv2 as cv
4 from google.colab.patches import cv2_imshow # for image display
5 from skimage import io
6 from PIL import Image
7 import matplotlib.pyplot as plt
```

2. Baca Gambar dari Url

Pada langkah ini kita akan membaca gambar dari url, dan menampilkannya menggunakan openCV, perhatikan perbedaannya saat membaca gambar dalam format RGB dan BGR. Saluran warna input default dalam format BGR untuk openCV.

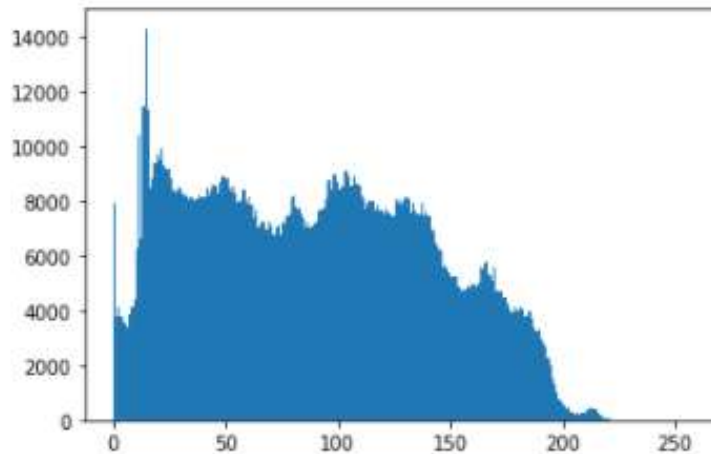
```
1 # Bagian yang berwarna hijau tidak usah diketik/ hanya komentar untuk penjelasan
2 # Buat daftar untuk menyimpan url gambar
3 urls = ["https://iiif.lib.ncsu.edu/iiif/0052574/full/800,/0/default.jpg",
4         "https://iiif.lib.ncsu.edu/iiif/0016007/full/800,/0/default.jpg",
5         "https://placekitten.com/800/571"]
6 # Baca dan tampilkan gambar
7 # Loop pada URL gambar, Anda dapat menyimpan beberapa url gambar dalam daftar
8 for url in urls:
9     image = io.imread(url)
10    image_2 = cv.cvtColor(image, cv.COLOR_BGR2RGB)
11    final_frame = cv.hconcat((image, image_2))
12    cv2_imshow(final_frame)
13    print('\n')
```

3. Kontur Gambar dan Histogram

```
1 # Cek tipe data matriks gambar (bisa mengetahui kedalaman bit gambar)
2 print(image.dtype)
3 # Cek ketinggian gambar
4 print(image.shape[0])
5 # Cek lebar gambar
6 print(image.shape[1])
7 # Cek jumlah saluran gambar
8 print(image.shape[2])
```

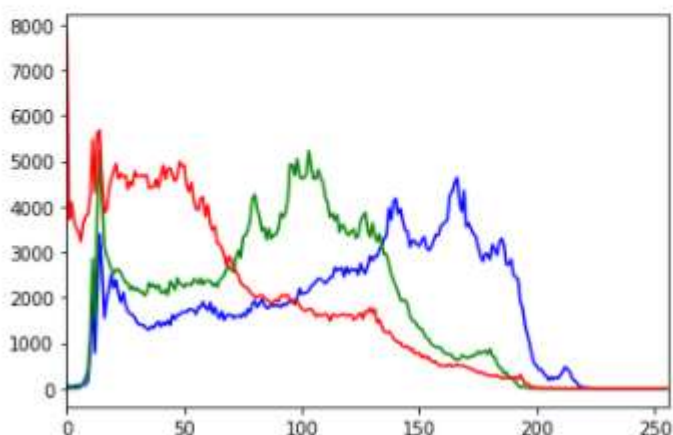
```
uint8
571
800
3
```

```
1 plt.hist(image.ravel(),bins = 256, range = [0,256])
2 plt.show()
```



Menampilkan histogram saluran R, G, B (Red, Green, Blue). Kita dapat mengamati bahwa saluran hijau memiliki banyak piksel di 255, yang mewakili warna putih pada gambar.

```
[5] 1 color = ('b','g','r')
    2 for i,col in enumerate(color):
    3     histr = cv.calcHist([image],[i],None,[256],[0,256])
    4     plt.plot(histr,color = col)
    5     plt.xlim([0,256])
    6 plt.show()
```



Untuk meningkatkan kontras pada gambar atau memperluas kontras di wilayah tertentu maka perlumengorbankan detail dalam warna yang tidak terlalu bervariasi, atau tidak penting.

Fungsi untuk menemukan area gambar yang menarik adalah histogram. Untuk membuat histogram dari data gambar menggunakan fungsi `matplotlib.pyplot.hist()`.

4. Menghasilkan Citra Grayscale

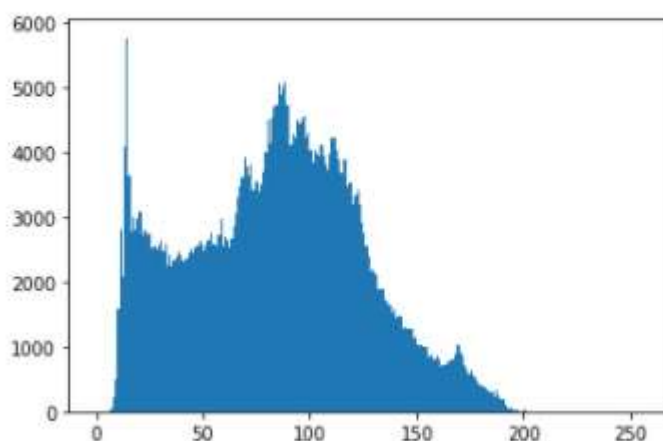
```
1 gray_image = cv.cvtColor(image, cv.COLOR_BGR2GRAY)
2 cv2.imshow(gray_image)
```



5. Plot Histogram

Plot histogram gambar abu-abu. Frekuensi histogram citra berkurang $\sim 1/3$ dari histogram citra berwarna.

```
1 plt.hist(gray_image.ravel(), bins = 256, range = [0, 256])
2 plt.show()
```



BAGIAN 2. Grayscale, Threshold, Flipping (Latihan2b_NIM.ipynb)

1. Perbedaan gambar RGB dan Grayscale

Apa perbedaan utama antara gambar yang ditunjukkan di bawah ini?



```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from skimage import data, color
4 from skimage.filters import threshold_otsu
5 from skimage.filters import try_all_threshold
6
```

```
[2] 1 def show_image(image, title='Image', cmap_type='gray'):
      2     plt.imshow(image, cmap=cmap_type)
      3     plt.title(title)
      4     plt.axis('off')
```

2. Memuat / load gambar dari modul data scikit-image

<https://scikit-image.org/docs/0.13.x/api/skimage.data.html>

```
1 from skimage import data
2
3 coffee_image = data.coffee()
4 coins_image = data.coins()
5 # Angka 3 menunjukkan ada 3 channel/saluran yaitu(Red,Green,Blue)
6 coffee_image.shape
```

(400, 600, 3)

```
1 # Sedangkan pada gambar coin hanya ada 1 channel yaitu Grayscale
2 coins_image.shape
```

(303, 384)

3. RGB to Grayscale

```
1 # Load citra rocket
2 rocket = data.rocket()
3
4 # Convert citra menjadi grayscale
5 gray_scaled_rocket = color.rgb2gray(rocket)
6
7 # Menampilkan original image
8 show_image(rocket, 'Original RGB image');
9
10
```

Original RGB image

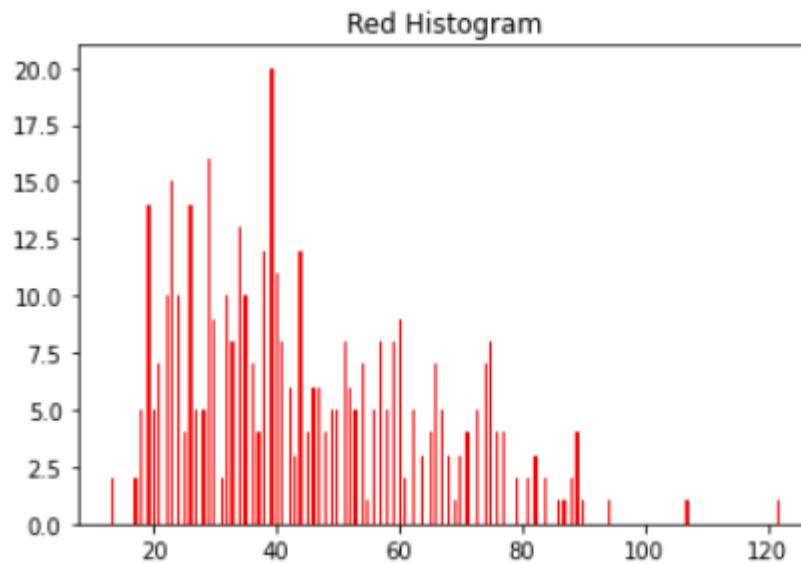


```
1 # Menampilkan citra grayscale
2 show_image(gray_scaled_rocket, 'Grayscale image')
```

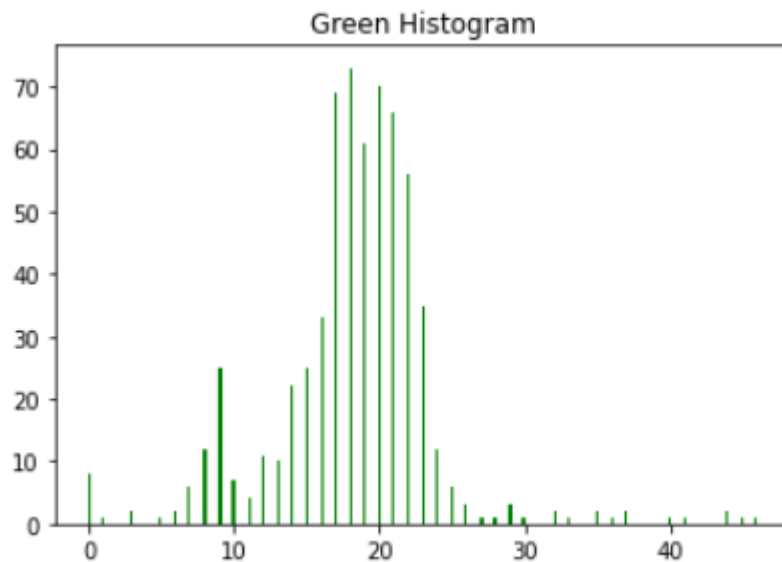
Grayscale image



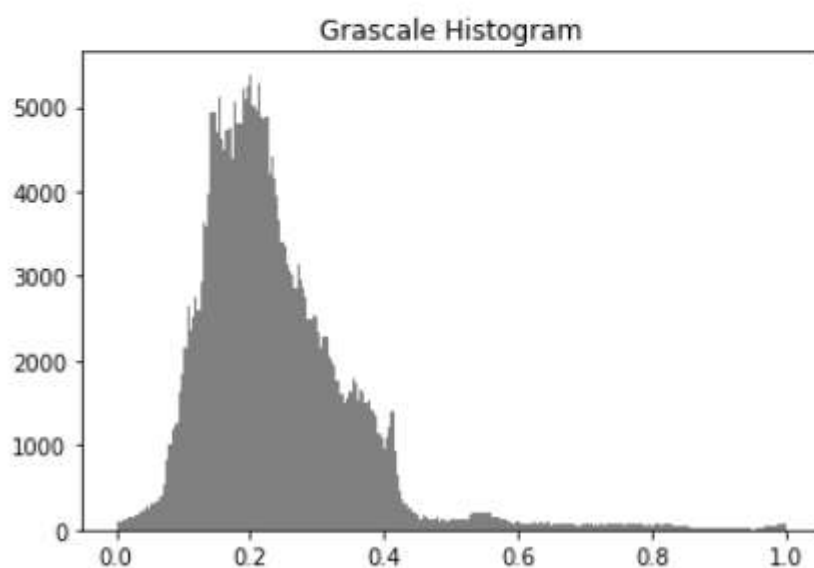
```
[24] 1 # Menampilkan histogram red channel
      2 red_channel = rocket[:, 0, 0]
      3
      4 # Plot the the red histogram with bins in a range of 256
      5 plt.hist(red_channel.ravel(), bins=256, color='red');
      6
      7 # Set title
      8 plt.title('Red Histogram');
```



```
[26] 1 # Menampilkan histogram green channel
      2 red_channel = rocket[0, :, 0]
      3
      4 # Plot the the red histogram with bins in a range of 256
      5 plt.hist(red_channel.ravel(), bins=256, color='green');
      6
      7 # Set title
      8 plt.title('Green Histogram');
```



```
1 # Menampilkan histogram Grayscale channel
2
3 # Plot the the grayscale histogram with bins in a range of 256
4 plt.hist(gray_scaled_rocket.ravel(), bins=256, color='gray');
5
6 # Set title
7 plt.title('Grascale Histogram');
```



4. Menerapkan Metode Thresholding

```
1 logo = data.logo()  
2 show_image(logo, 'Citra Original');
```



Citra Original



```
1 # Mengubah citra grayscale menggunakan fungsi rgb2gray  
2 logo_gray = color.rgb2gray(logo)  
3  
4 # Dapatkan nilai ambang optimal dengan metode otsu  
5 thresh = threshold_otsu(logo_gray)  
6  
7 # Apply thresholding ke citra  
8 binary = logo_gray > thresh  
9  
10 # menampilkan citra  
11 show_image(binary, 'Binary image')
```

Binary image



```
1 # Menerapkan semua metode Threshold pada citra grayscale  
2 fig, ax = try_all_threshold(logo_gray, verbose=False);
```

Original



Li



Minimum



Triangle



Isodata



Mean



Otsu



Yen



5. Menerapkan Flipping

```
1 cat = data.chelsea()  
2 show_image(cat, 'Citra Original');  
3
```

Citra Original



```
1 # Mengubah citra grayscale menggunakan fungsi rgb2gray  
2 cat_gray = color.rgb2gray(cat)  
3 show_image(cat_gray, 'Citra Grayscale');
```

Citra Grayscale



```
1 # Flip dari kiri ke kanan ( fliplr() )  
2 cat_horizontal_flip = np.fliplr(cat_gray)  
3 show_image(cat_horizontal_flip, 'Cat Horizontal')
```

Cat Horizontal



BAGIAN 3. Rotasi, Resize/Rescale Image

1. Import library



```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 import cv2
4 from skimage import data, color
5 from skimage.transform import rotate
6 from skimage.color import rgb2gray
```

2. Upload gambar lenna.jpg

```
[7] 1 from google.colab import files
    2 uploaded = files.upload()
```

Choose Files lenna.JPG

- lenna.JPG(image/jpeg) - 51779 bytes, last modified: 4/19/2021 - 100% done
Saving lenna.JPG to lenna (1).JPG

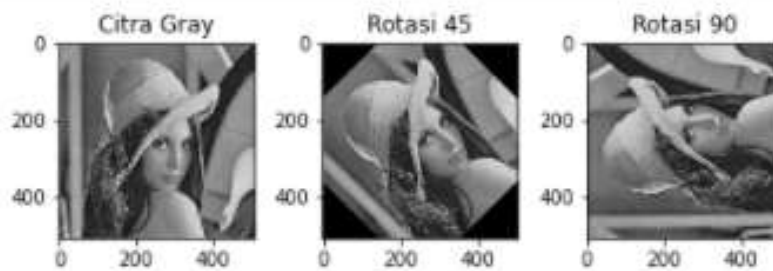
3. Membaca gambar lenna dan konversi ke grayscale

```
1 from skimage.io import imread, imshow
2
3 %matplotlib inline
4
5 img = imread('lenna.JPG')
6 citra_gray = rgb2gray(img)
7 plt.subplot(121), imshow(img)
8 plt.title('RGB Format')
9
10 plt.subplot(122), imshow(citra_gray)
11 plt.title('Grayscale Format')
12
13 plt.show()
```



4. Rotasi Citra

```
1 plt.subplot(131), imshow(citra_gray)
2 plt.title('Citra Gray')
3 rotasi45 = rotate(citra_gray, angle=45)
4 plt.subplot(132), imshow(rotasi45)
5 plt.title('Rotasi 45')
6 rotasi90 = rotate(citra_gray, angle=90)
7 plt.subplot(133), imshow(rotasi90)
8 plt.title('Rotasi 90')
9
10 plt.show()
```



5. Rezise/Rescale Citra

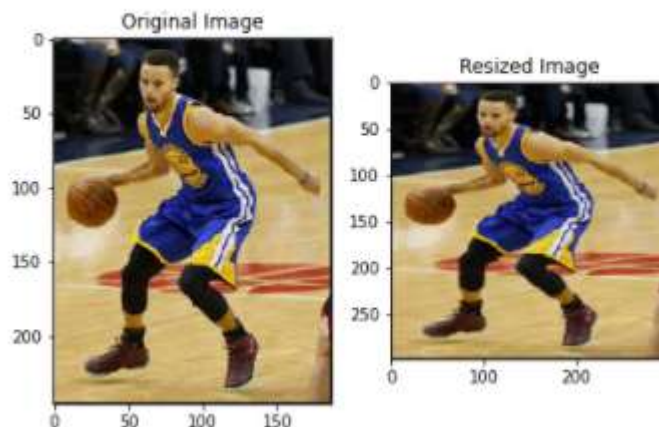
Citra harus diubah ke Grayscale agar channel menjadi 1 yaitu grayscale dan bukan 3 channel / RGB

```
[65] 1 from google.colab import files
      2 uploaded = files.upload()
```

Choose Files ball.JPG

- ball.JPG(image/jpeg) - 20434 bytes, last modified: 4/19/2021 - 100% done
Saving ball.JPG to ball.JPG

```
[67] 1 from skimage.transform import resize
      2 img = imread('ball.JPG')
      3 #resize image
      4 img_resized = resize(img, (300, 300))
      5
      6 #plot images
      7 plt.subplot(121), imshow(img)
      8 plt.title('Original Image')
      9 plt.subplot(122), imshow(img_resized)
     10 plt.title('Resized Image')
     11 plt.show()
```



```
1 from skimage.transform import rescale
2 gbr = imread('ball.JPG')
3 gbr_gray=rgb2gray(gbr)
4
5 gbr2 = rescale(gbr_gray, scale=2)
6
7 plt.subplot(121), imshow(gbr_gray)
8 plt.title('Original Image')
9
10 plt.subplot(122), imshow(gbr2)
11 plt.title('Rescaled Image')
12
13 plt.show()
```

