1. Jelaskan step2 thresholding dan detail proses opencv diatas dan apa perbedaan/kesamaannya.

---

2. Tambahkan masing masing histogram hasil thresholding (jika tidak bisa jelaskan)

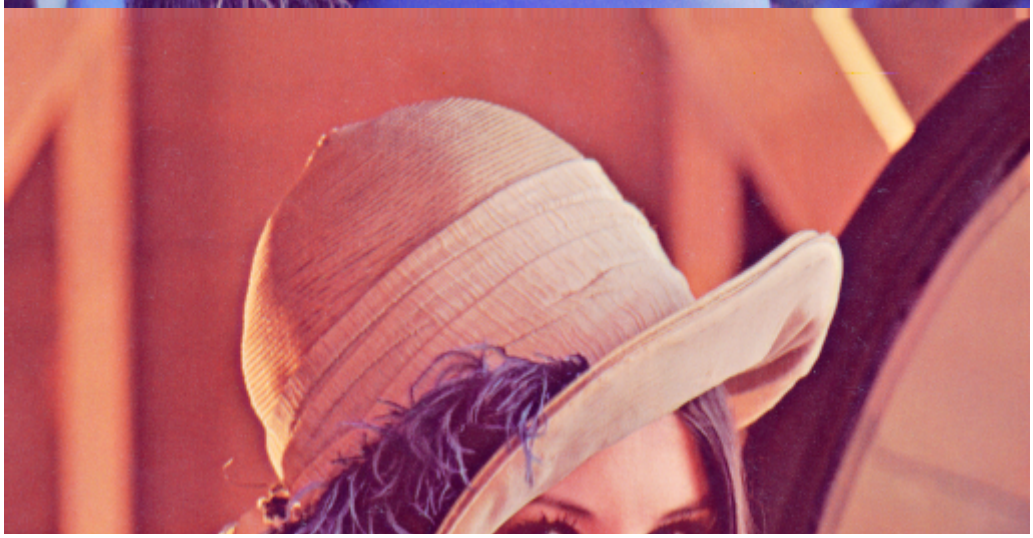
---

3. tampilkan histogram RGB dari https://drive.google.com/file/d/1eWVCSyFKtCbDT_nv2rN3vce4HD-2Pr3h/view?usp=sharing
4. Buat fungsi untuk melakukan histogram equalization dengan masukan sebuah image(jelaskan pdf dan cdf) https://www.math.uci.edu/icamp/courses/math77c/demos/hist_eq.pdf

---

5. Buat fungsi untuk melakukan Gamma Correction dengan masukan sebuah image (foto diri anda di Lab MM)

```
import numpy as np
import pandas as pd
import cv2 as cv
from google.colab.patches import cv2_imshow # for image display
from skimage import io
from PIL import Image
import matplotlib.pylab as plt


url='https://upload.wikimedia.org/wikipedia/en/7/7d/Lenna_%28test_image%29.png'
image_0 = io.imread(url)
image_2 = cv.cvtColor(image_0, cv.COLOR_BGR2RGB)

cv2_imshow(image_0)
cv2_imshow(image_2)
```



```
### TEORI
Thresholding merupakan sebuah proses membiner
Sederhananya, sebuah gambar RGB dapat diubah
menentukan nilai thresholding T, kemudian mis
sebagai nilai piksel sebagai dua kelompok, mi



### IMPLEMENTASI
Dalam function thresholding2(), menerima para

Terdapat beberapa langkah yang dilakukan, yak
1. Melakukan konversi **color format** gambar
method `cv.cvtColor()`
```

TEORI

Thresholding merupakan sebuah proses membinerisasi (binarization) gambar. Sederhananya, sebuah gambar RGB dapat diubah ke dalam bentuk biner dengan menentukan nilai thresholding T, kemudian misalnya merepsentasikan biner sebagai nilai piksel sebagai dua kelompok, misalnya p < T dan P >= T.

```
...method = cv.cvtColor()

2. Melakukan proses blurring menggunakan meth
kernel size (5,5) serta kernel standar deviat
```
```
blur = cv.GaussianBlur(img, (5,5), 0)
```
```
3. Melakukan proses thresholding menggunakan
argumen pertama input gambar yang sudah di bl
threshold (T) 0, hal ini karena nantinya akar
mengkomputasi nilai optimal thresholding T. P
output piksel jika melewati thresholding test
teknik yang digunakan untuk melakukan thresho
THRESH_BINARY_INV` disertai extra flag `cv2.T
` akan me-return dua value, yaitu threshold T
threshold dari teknik Otsu.
```
```
ret2, thresh = cv.threshold(blur, 0, 255, cv.
```
```

4. Melakukan proses thresholding berjenis *G
threshold T sebesar 127. Menggunakan teknik `
nilai piksel melebihi nilai T, maka akan diub
ketiga yitu 255.

5. Melakukan proses thresholding berjenis *Ad
menggunakan method `cv2.adaptiveThreshold`. D
adalah rata-rata dari area tetangga dikurangi

6. Melakukan proses thresholding berjenis *Ad
Dengan teknik ini, nilai threshold adalah jum
tetangga dikurangi konstanta C

*Adaptive Thresholding* dibuat untuk mengatas
Thresholding* dimana nilai threshold T yang s
Padahal terdapat beberapa faktor yang dapat m
seperti pencahayaan, shadow, dll. Dengan itu,
berlaku di bagian tertentu saja dalam gambar.

*Adaptive Thresholding* mempertimbangkan satu
suatu waktu, menghitung nilai T untuk bagian
melakukan segmentasi.
```

IMPLEMENTASI

Dalam function thresholding2(), menerima parameter berupa `image`

Terdapat beberapa langkah yang dilakukan, yakni:

1. Melakukan konversi **color format** gambar dari BGR menjadi Grayscale dengan method `cv.cvtColor()`
2. Melakukan proses blurring menggunakan method `cv.GaussianBlur()` dengan kernel size (5,5) serta kernel standar deviation sebesar 0

```
blur = cv.GaussianBlur(img, (5,5), 0
```

3. Melakukan proses thresholding menggunakan method `cv.threshold()` dengan argumen pertama input gambar yang sudah di blur. Pada argumen kedua terdapat threshold (T) 0, hal ini karena nantinya akan digunakan flag teknik Otsu untuk mengkomputasi nilai optimal thresholding T. Pada argumen ketiga, berisi nilai output piksel jika melewati thresholding test. Pada argumen keempat, berisi teknik yang digunakan untuk melakukan thresholding yaitu `cv2.THRESH_BINARY_INV` disertai extra flag `cv2.THRESH_OTSU`. Method `cv.threshold()` akan me-return dua value, yaitu threshold T dan threshold gambar (optimal threshold dari teknik Otsu.

```
ret2, thresh = cv.threshold(blur, 0,
```

4. Melakukan proses thresholding berjenis *Global Thresholding* dengan nilai threshold T sebesar 127. Menggunakan teknik `cv.THRESH_BINARY` sehingga jika nilai piksel melebihi nilai T, maka akan diubah ke nilai output pada argumen ketiga yitu 255.
5. Melakukan proses thresholding berjenis *Adaptive Mean Thresholding* menggunakan method `cv2.adaptiveThreshold`. Dengan teknik ini, nilai threshold adalah rata-rata dari area tetangga dikurangi konstanta C
6. Melakukan proses thresholding berjenis *Adaptive Gaussian Thresholding*. Dengan teknik ini, nilai threshold adalah jumlah *gaussian-weighted* dari nilai tetangga dikurangi konstanta C

Adaptive Thresholding dibuat untuk mengatasi masalah yang ada pada *Simple Thresholding* dimana nilai threshold T yang sama digunakan untuk piksel gambar. Padahal terdapat beberapa faktor yang dapat mengganggu proses thresholding, seperti pencahayaan, shadow, dll. Dengan itu, mungkin suatu threshold T hanya berlaku di bagian tertentu saja dalam gambar.

Adaptive Thresholding mempertimbangkan satu set kecil piksel tetangga pada suatu waktu, menghitung nilai T untuk bagian lokal tertentu, dan kemudian melakukan segmentasi.

Nomor 1 & 2

[ ] ↳ *2 sel tersembunyi*

Nomor 3

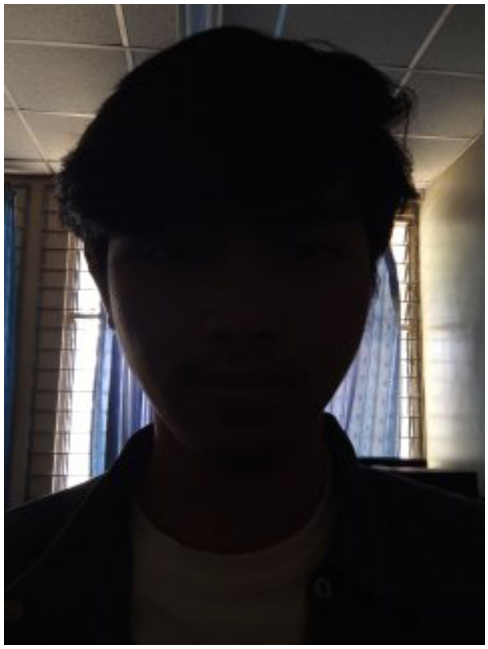
[ ] ↳ *3 sel tersembunyi*

Nomor 4

[ ] ↳ *6 sel tersembunyi*

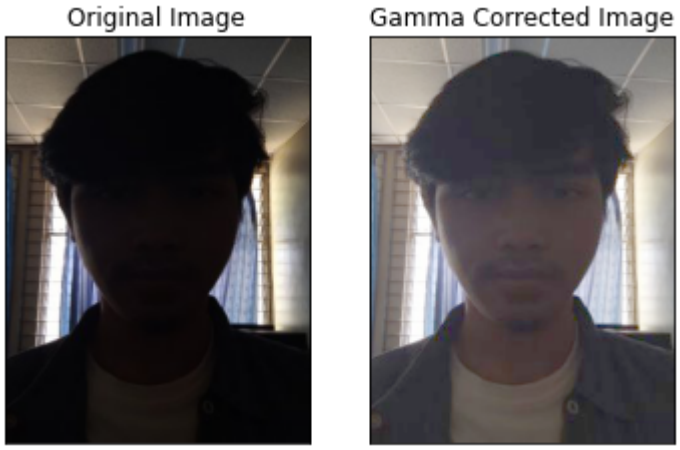
Nomor 5

```
url_5 ='https://raw.githubusercontent.com/wajisobri/image-folder/master/selfie1.jpg'
image_5 = io.imread(url_5)

url_6 ='https://raw.githubusercontent.com/wajisobri/image-folder/master/selfie2.jpg'
image_6 = io.imread(url_6)

cv2_imshow(cv.cvtColor(image_5, cv.COLOR_BGR2RGB))
```

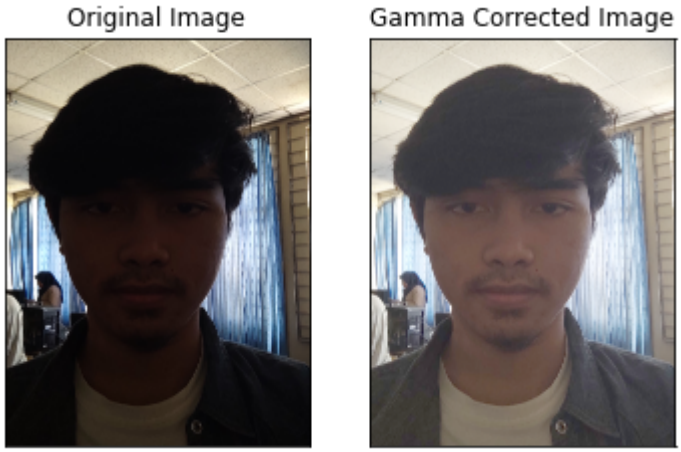


```
def gamma_correction(image, gamma):
    exponent = 1 / gamma

    table = [((i / 255) ** exponent) * 255 for i in range(256)]
    table = np.array(table, np.uint8)

    plt.subplot(1, 2, 1), plt.imshow(image)
    plt.title('Original Image')
    plt.xticks([]), plt.yticks([])

    plt.subplot(1, 2, 2), plt.imshow(cv.LUT(image, table))
    plt.title('Gamma Corrected Image')
    plt.xticks([]), plt.yticks([])
```

```
gamma_correction(image_5, 2.4)
```




```
gamma_correction(image_6, 2.4)
```



Referensi

https://pyimagesearch.com/2021/05/12/adaptive-thresholding-with-opencv-cv2-adaptivethreshold/ https://pyimagesearch.com/2021/04/28/opencv-thresholding-cv2-threshold/
https://www.bogotobogo.com/python/OpenCV_Python/python_opencv3_Image_Global_Thresholding_Adaptive_Thresholding_Otsus_Binarization_Segmentations.php
https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html
https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html https://lindevs.com/apply-gamma-correction-to-an-image-using-opencv

Produk berbayar Colab - Batalkan kontrak di sini

✓ 5 d selesai pada 22.53