

---

# HISTOGRAM EQUALIZATION , DITHERING

## MODULE 6

---



Arranged By:

Rajendra Rakha Arya Prabaswara

(1941720080/20)

PROGRAM STUDI D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG



**1. LINK GITHUB**

[https://github.com/Rindrakha/PCVK\\_Genap\\_2022](https://github.com/Rindrakha/PCVK_Genap_2022)

**2. Mount Image From Drive**

```
+ Kode + Teks Salin ke Drive

from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

2. Import beberapa library berikut yang akan digunakan selama uji coba praktikum minggu ke-6 berikut

[ ] import cv2 as cv
    from google.colab.patches import cv2_imshow
    from skimage import io
    import matplotlib.pyplot as plt
    import numpy as np
    import math
    import os
    import glob
```

**3. Make an image histogram like the following histogram output based on the flowchart below : (Use image lena.jpg)**



4. After working on question no. 3, Compare the results. Does the output appear the same?



5. Make an image histogram such as the output equalization histogram and also the image display before and after the following equalization histogram process based on the flowchart below : (Use image lena\_lc.jpg)

```

img = cv.imread('/content/drive/MyDrive/PCVK_MM/Images/lena_lc.jpg')
def histog(img):
    height, width, depth = np.shape(img)
    ax = np.zeros((3,256))
    for y in range(0,height):
        for x in range(0,width):
            ax[0][img[y][x][0]] += 1
            ax[1][img[y][x][1]] += 1
            ax[2][img[y][x][2]] += 1
    return ax

histo_asli = histog(img)

#convert to Numpy array
img_array = np.asarray(img)

#STEP 1: Normalized cumulative histogram
#flatten image array and calculate histogram via binning
histogram_array = np.bincount(img_array.flatten(), minlength=256)

#normalize
num_pixels = np.sum(histogram_array)
histogram_array = histogram_array/num_pixels

#normalized cumulative histogram
chistogram_array = np.cumsum(histogram_array)

#STEP 2: Pixel mapping lookup table
transform_map = np.floor(255 * chistogram_array).astype(np.uint8)

#STEP 3: Transformation
# flatten image array into 1D list
img_list = list(img_array.flatten())

# transform pixel values to equalize
eq_img_list = [transform_map[p] for p in img_list]

# reshape and write back into img_array
eq_img_array = np.reshape(np.asarray(eq_img_list), img_array.shape)
img2 = eq_img_array

equal_histo = histog(img2)

final_frame = cv.hconcat((img, img2)) #concatenate image
cv2.imshow('final_frame')

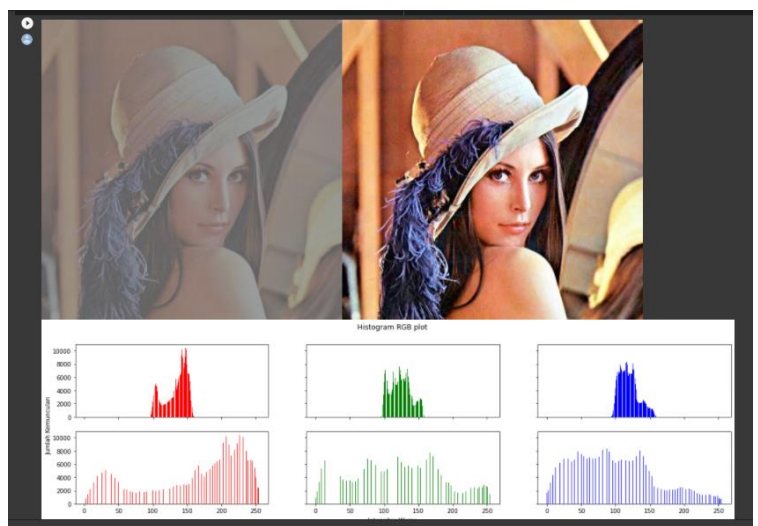
```

```

final_frame = cv.hconcat((img, img2)) #concatenate image
cv2.imshow('final_frame')

names = np.arange(256)
isgrayimg = np.array_equal(blue, green, red)
if (isgrayimg == False):
    fig, axes = plt.subplots(2, 3, figsize=[20,5], sharex=True, sharey=True)
    fig.suptitle('Histogram RGB plot')
    fig.text(0.09, 0.5, 'Jumlah Kemunculan', va='center', rotation='vertical')
    fig.text(0.5, 0.04, 'Intensitas Warna', ha='center')
    axes[0,0].bar(names, histo_asli[2], color='red')
    axes[0,1].bar(names, histo_asli[1], color='green')
    axes[0,2].bar(names, histo_asli[0], color='blue')
    axes[1,0].bar(names, equal_histo[2], color='red')
    axes[1,1].bar(names, equal_histo[1], color='green')
    axes[1,2].bar(names, equal_histo[0], color='blue')
else:
    fig, axes = plt.subplots(1, 2, figsize=[20,5], sharex=True, sharey=True)
    fig.suptitle('Histogram RGB plot')
    fig.text(0.09, 0.5, 'Jumlah Kemunculan', va='center', rotation='vertical')
    fig.text(0.5, 0.04, 'Intensitas Warna', ha='center')
    axes[0,0].bar(names, histo_asli[2], color='gray')
    axes[0,1].bar(names, equal_histo[2], color='gray')

```



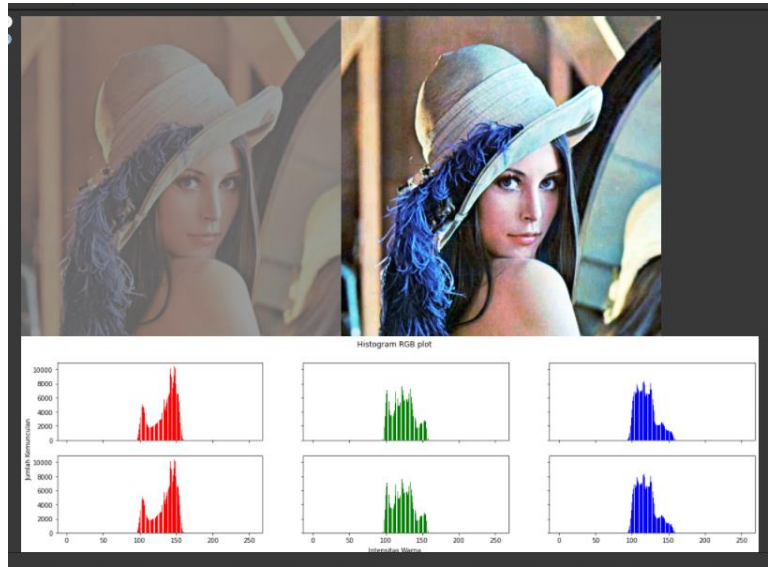


6. After working on question no. 5, make the same image histogram but use the library owned by CV2, namely "equalizeHist". Compare the results. Does the output appear the same?

```
img = cv.imread('/content/drive/MyDrive/PCVK_MM/Images/lena_lc.jpg')
channels = cv.split(img)
eq_channels = []
for ch, color in zip(channels, ['R', 'G', 'B']):
    eq_channels.append(cv.equalizeHist(ch))
eq_image = cv.merge(eq_channels)
equal_histo = histog(eq_image)

final_frame = cv.hconcat((img, eq_image))
cv2.imshow('final_frame')

names = np.arange(256)
isgraying = np.array_equal(blue, green, red)
if (isgraying == False):
    fig, axs = plt.subplots(2, 3, figsize=[20,5], sharex=True, sharey=True)
    fig.suptitle('Histogram RGB plot')
    fig.text(0.09, 0.5, 'Jumlah Kemunculan', va='center', rotation='vertical')
    fig.text(0.5, 0.04, 'Intensitas Warna', ha='center')
    axs[0,0].bar(names, histo_asli[2], color='red')
    axs[0,1].bar(names, histo_asli[1], color='green')
    axs[0,2].bar(names, histo_asli[0], color='blue')
    axs[1,0].bar(names, histo_asli[2], color='red')
    axs[1,1].bar(names, histo_asli[1], color='green')
    axs[1,2].bar(names, histo_asli[0], color='blue')
else:
    fig, axs = plt.subplots(1, 2, figsize=[20,5], sharex=True, sharey=True)
    fig.suptitle('Histogram RGB plot')
    fig.text(0.09, 0.5, 'Jumlah Kemunculan', va='center', rotation='vertical')
    fig.text(0.5, 0.04, 'Intensitas Warna', ha='center')
    axs[0,0].bar(names, histo_asli[2], color='gray')
    axs[0,1].bar(names, equal_histo[2], color='gray')
```



7. Do the Floyd and Steinberg dithering process like the following output (initial image display, and after dithering display) based on the flowchart below! (Use wiki.jpg image)

```
#No 8
from PIL import Image
import cv2
import numpy as np

imageConvert = Image.open('/content/drive/MyDrive/PCVK/Images/wiki.jpg').convert(mode='1', dither=Image.FLOYDSTEINBERG)
imageConvert.save('/content/drive/MyDrive/PCVK/Images/wiki.jpg')

Image = cv2.imread('/content/drive/MyDrive/PCVK/Images/wiki.jpg')
GrayImage = cv2.cvtColor(Image, cv2.COLOR_BGR2GRAY)
cv2.imwrite('/content/drive/MyDrive/PCVK/Images/wiki.jpg', GrayImage)

Height = GrayImage.shape[0]
Width = GrayImage.shape[1]

for y in range(0, Height):
    for x in range(0, Width):
        old_value = GrayImage[y, x]
        new_value = 0
        if (old_value > 128) :
            new_value = 255
        GrayImage[y, x] = new_value
        Error = old_value - new_value

        if (x<Width-1):
            NewNumber = GrayImage[y, x+1] + Error * 7 / 16
            if (NewNumber>255) : NewNumber=255
            elif (NewNumber<0) : NewNumber=0
            GrayImage[y, x+1] = NewNumber

        if (x<0 and y<Height-1):
            NewNumber = GrayImage[y+1, x-1] + Error * 3 / 16
            if (NewNumber>255) : NewNumber=255
            elif (NewNumber<0) : NewNumber=0
            GrayImage[y+1, x-1] = NewNumber

        if (y<Height-1):
            NewNumber = GrayImage[y+1, x] + Error * 5 / 16
            if (NewNumber>255) : NewNumber=255
            elif (NewNumber<0) : NewNumber=0
            GrayImage[y+1, x] = NewNumber

        if (y<Height-1 and x<Width-1):
            NewNumber = GrayImage[y+1, x+1] + Error * 1 / 16
            if (NewNumber>255) : NewNumber=255
            elif (NewNumber<0) : NewNumber=0
            GrayImage[y+1, x+1] = NewNumber

cv2.imwrite('/content/drive/MyDrive/PCVK/Images/wiki.jpg', GrayImage)
cv2.imshow(GrayImage)
```





## **Jurusan Teknologi Informasi Politeknik Negeri Malang.**

Rajendra Rakha Arya Prabaswara

---