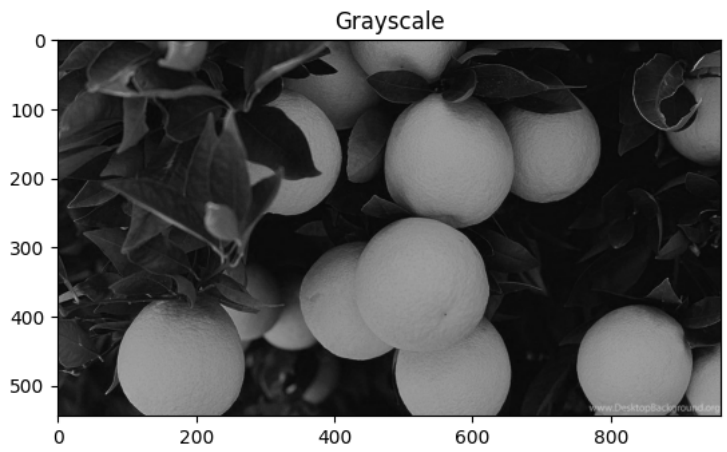
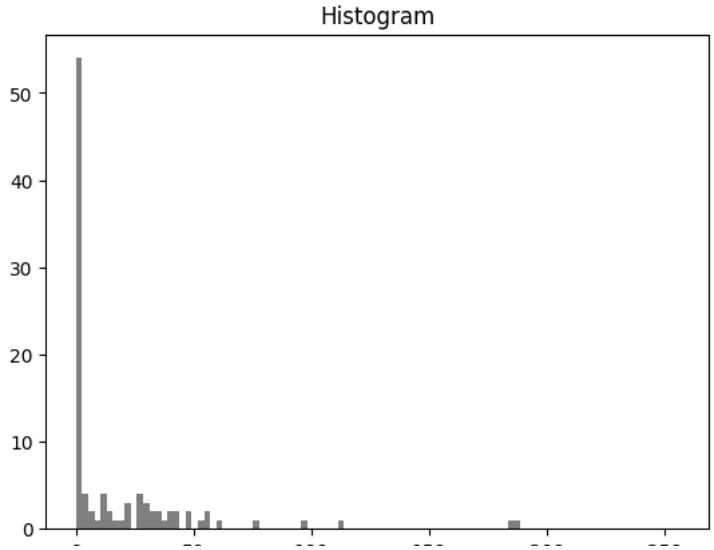
```
#ARDIANSYAH - 1207070018 - TEKNIK ELEKTRO (TSEB)
import numpy as np  # Mengimpor modul numpy untuk operasi array
import imageio  # Mengimpor modul imageio untuk membaca gambar
import matplotlib.pyplot as plt  # Mengimpor modul matplotlib.pyplot untuk plot grafik
from google.colab import drive  # Mengimpor modul google.colab.drive untuk mengakses Google Drive
drive.mount('/content/drive')  # Mengaitkan dan mengakses Google Drive

img = imageio.imread('/content/drive/MyDrive/Colab Notebooks/orange.jpg')  # Membaca gambar dengan menggunakan imageio
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
<ipython-input-2-43a95072c500>:7: DeprecationWarning: Starting with ImageIO v3 the behavior of this function will switch to that of
  img = imageio.imread('/content/drive/MyDrive/Colab Notebooks/orange.jpg')
```

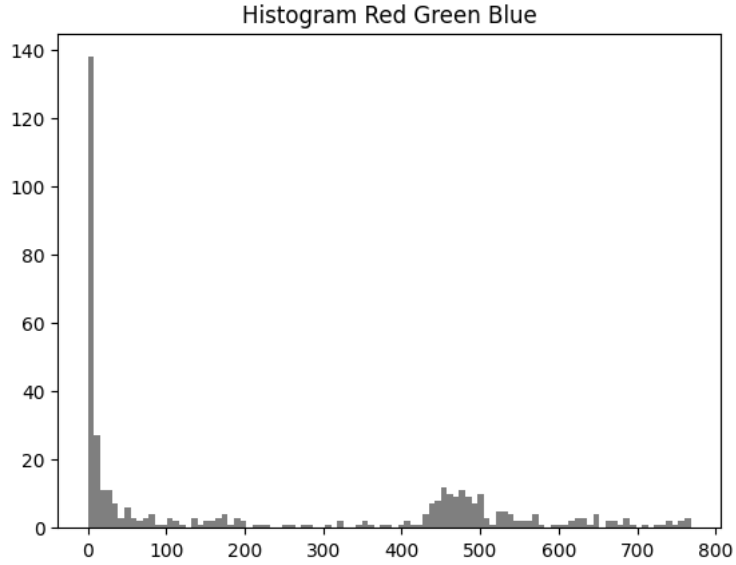
```
img_height = img.shape[0]  # Mengambil tinggi gambar
img_width = img.shape[1]  # Mengambil lebar gambar
img_channel = img.shape[2]  # Mengambil jumlah saluran warna gambar


img_grayscale = np.zeros(img.shape, dtype=np.uint8)  # Membuat array kosong dengan ukuran yang sama dengan gambar untuk menyimpan citra c

for y in range(0, img_height):  # Melakukan loop untuk setiap baris dalam gambar
    for x in range(0, img_width):  # Melakukan loop untuk setiap piksel dalam baris
        red = img[y][x][0]  # Mengambil nilai warna merah piksel
        green = img[y][x][1]  # Mengambil nilai warna hijau piksel
        blue = img[y][x][2]  # Mengambil nilai warna biru piksel
        gray = (int(red) + int(green) + int(blue)) / 3  # Menghitung nilai keabuan dari rata-rata nilai warna piksel
        img_grayscale[y][x] = (gray, gray, gray)  # Menyimpan nilai keabuan dalam array citra skala keabuan

plt.imshow(img_grayscale)  # Menampilkan gambar dalam skala keabuan
plt.title("Grayscale")  # Menambahkan judul pada plot
plt.show()  # Menampilkan plot
```



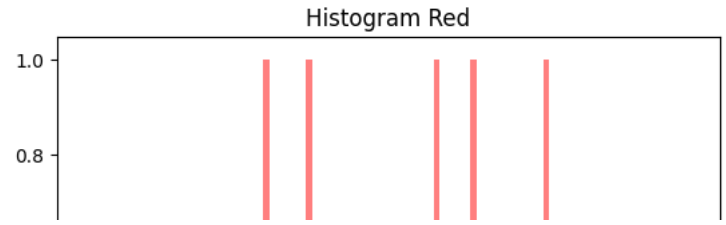

```
hg = np.zeros((256))  # Membuat array kosong dengan ukuran 256 untuk menyimpan histogram


for x in range(0, 256):  # Melakukan loop untuk setiap nilai histogram
    hg[x] = 0  # Menginisialisasi nilai histogram awal


for y in range(0, img_height):  # Melakukan loop untuk setiap baris dalam gambar
    for x in range(0, img_width):  # Melakukan loop untuk setiap piksel dalam baris
        gray = img_grayscale[y][x][0]  # Mengambil nilai keabuan piksel
        hg[gray] += 1  # Menghitung frekuensi kemunculan nilai keabuan dalam histogram


# plt.figure(figsize=(20, 6))
# plt.plot(hg, color="black", linewidth=2.0)
# plt.show()

bins = np.linspace(0, 256, 100)  # Membuat daftar interval untuk histogram dengan 100 elemen dari 0 hingga 256
plt.hist(hg, bins, color="black", alpha=0.5)  # Menampilkan histogram dengan data 'hg', menggunakan interval 'bins', berwarna hitam, dan
plt.title("Histogram")  # Menambahkan judul pada gambar histogram
plt.show()  # Menampilkan gambar histogram
```



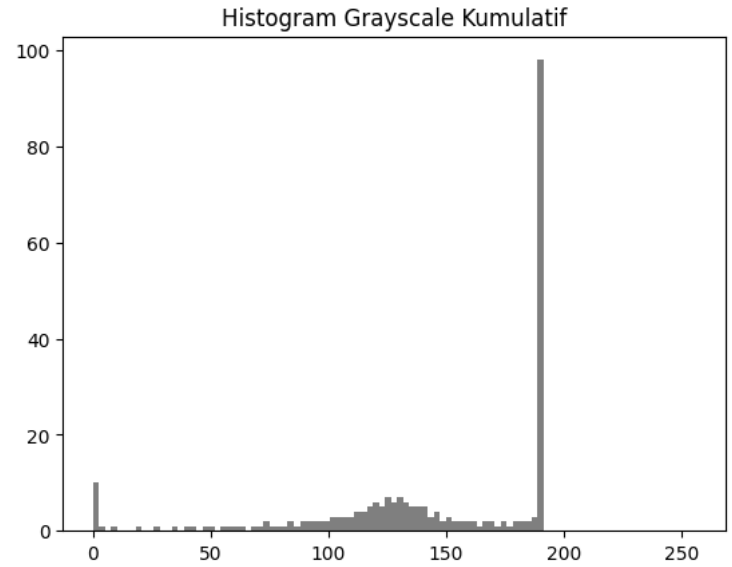

```
hgr = np.zeros((256))  # Membuat array kosong dengan ukuran 256 untuk nilai histogram merah
hgg = np.zeros((256))  # Membuat array kosong dengan ukuran 256 untuk nilai histogram hijau
hgb = np.zeros((256))  # Membuat array kosong dengan ukuran 256 untuk nilai histogram biru
hgrgb = np.zeros((768))  # Membuat array kosong dengan ukuran 768 untuk nilai histogram merah, hijau, dan biru


for x in range(0, 256):
    hgr[x] = 0  # Menginisialisasi array hgr dengan nilai 0
    hgg[x] = 0  # Menginisialisasi array hgg dengan nilai 0
    hgb[x] = 0  # Menginisialisasi array hgb dengan nilai 0

for x in range(0, 768):
    hgrgb[x] = 0  # Menginisialisasi array hgrgb dengan nilai 0


# th = int(256/64)
temp = [0]
for y in range(0, img.shape[0]):
    for x in range(0, img.shape[1]):
        red = int(img[y][x][0])
        green = int(img[y][x][1])
        blue = int(img[y][x][2])
        green = green + 256
        blue = blue + 512
#         temp.append(green)
        hgrgb[red] += 1  # Menambahkan nilai pada array hgrgb untuk red
        hgrgb[green] += 1  # Menambahkan nilai pada array hgrgb untuk green
        hgrgb[blue] += 1  # Menambahkan nilai pada array hgrgb untuk blue

binsrgb = np.linspace(0, 768, 100)
plt.hist(hgrgb, binsrgb, color="black", alpha=0.5)
# plt.plot(hgrgb)
plt.title("Histogram Red Green Blue")
plt.show()
```



```
    for y in range(0, img_height):
        for x in range(0, img_width):
            red = img[y][x][0]
            green = img[y][x][1]
            blue = img[y][x][2]
            hgr[red] += 1  # Menambahkan nilai pada array hgr untuk red
            hgg[green] += 1  # Menambahkan nilai pada array hgg untuk green
            hgb[blue] += 1  # Menambahkan nilai pada array hgb untuk blue

    bins = np.linspace(0, 256, 100)
    plt.hist(hgr, bins, color="red", alpha=0.5)
    plt.title("Histogram Red")
    plt.show()

    plt.hist(hgg, bins, color="green", alpha=0.5)
    plt.title("Histogram Green")
    plt.show()

    plt.hist(hgb, bins, color="blue", alpha=0.5)
    plt.title("Histogram Blue")
    plt.show()
```




```
# Membuat array kosong dengan ukuran 256 untuk variabel hgk, c, hgh, dan h
hgk = np.zeros((256))
c = np.zeros((256))
hgh = np.zeros((256))
h = np.zeros((256))

# Mengisi array hgk dan c dengan nilai 0
for x in range(0, 256):
    hgk[x] = 0
    c[x] = 0

# Looping untuk menghitung histogram grayscale kumulatif
# pada setiap pixel gambar (img_grayscale)
for y in range(0, img_height):
    for x in range(0, img_width):
        gray = img_grayscale[y][x][0]
        hgk[gray] += 1

# Mengisi nilai c[0] dengan nilai hgk[0]
c[0] = hgk[0]

# Menghitung nilai kumulatif c[x] dengan menggunakan nilai c[x-1] dan hgk[x]
for x in range(1, 256):
     c[x] = c[x-1] + hgk[x]

# Menghitung nilai maksimum hmaxk dari c[255]
hmaxk = c[255]

# Mengubah nilai c[x] dengan mengalikannya dengan 190 dan membaginya dengan hmaxk
for x in range(0, 256):
    c[x] = 190 * c[x] / hmaxk

# Menampilkan histogram grayscale kumulatif menggunakan plt.hist
plt.hist(c, bins, color="black", alpha=0.5)
plt.title("Histogram Grayscale Kumulatif")
plt.show()
```




```
# Inisialisasi array hgh, h, dan c dengan nilai nol
hgh = np.zeros((256))
h = np.zeros((256))
c = np.zeros((256))

# Mengisi array hgh, h, dan c dengan nol
for x in range(0, 256):
    hgh[x] = 0
    h[x] = 0
    c[x] = 0
```

```
# Menghitung jumlah kemunculan tiap nilai gray pada gambar grayscale dan menyimpannya dalam array hgh
for y in range(0, img_height):
    for x in range(0, img_width):
        gray = img_grayscale[y][x][0]
        hgh[gray] += 1

# Menghitung kumulatif dari array hgh dan menyimpannya dalam array h
h[0] = hgh[0]
for x in range(1, 256):
     h[x] = h[x-1] + hgh[x]

# Normalisasi array h
for x in range(0, 256):
     h[x] = h[x] / img_height / img_width

# Mengosongkan array hgh
for x in range(0, 256):
    hgh[x] = 0

# Menghitung hasil equalisasi histogram dan menyimpannya dalam array hgh
for y in range(0, img_height):
    for x in range(0, img_width):
        gray = img_grayscale[y][x][0]
        gray = h[gray] * 255
        hgh[int(gray)] += 1

# Menghitung kumulatif dari array hgh dan menyimpannya dalam array c
c[0] = hgh[0]
for x in range(1, 256):
     c[x] = c[x-1] + hgh[x]

# Menentukan nilai maksimum dari array c
hmaxk = c[255]

# Melakukan normalisasi pada array c dengan faktor 190 dan hmaxk
for x in range(0, 256):
    c[x] = 190 * c[x] / hmaxk

# Menampilkan histogram menggunakan array c sebagai data dan menentukan bin dan warna histogram
plt.hist(c, bins, color="black", alpha=0.5)
plt.title("Histogram Grayscale Hequalisasi")
plt.show()
```