

```

#ARDIANSYAN - 1207070018 - TEKNIK ELEKTRO (TSEB)
import numpy as np # Mengimpor pustaka NumPy untuk operasi array dan matriks.
import imageio # Mengimpor pustaka imageio untuk membaca dan menulis gambar.
import matplotlib.pyplot as plt # Mengimpor pustaka matplotlib untuk visualisasi gambar.
from google.colab import drive # Mengimpor pustaka google.colab untuk menghubungkan dan mengakses Google Drive.
drive.mount('/content/drive') # Mount Google Drive ke Colab.

img = imageio.imread('/content/drive/MyDrive/Colab Notebooks/orange.jpg') # Membaca gambar dari path tertentu.

Mounted at /content/drive
<ipython-input-1-ce9704413277>:7: DeprecationWarning: Starting with ImageIO v3 the behavior of this function will switch to that of iio.v3.imread. To keep the current behavior (and make this warn
img = imageio.imread('/content/drive/MyDrive/Colab Notebooks/orange.jpg') # Membaca gambar dari path tertentu.

img_height = img.shape[0] # Mendapatkan tinggi gambar dalam piksel.
img_width = img.shape[1] # Mendapatkan lebar gambar dalam piksel.
img_channel = img.shape[2] # Mendapatkan jumlah saluran warna gambar.

img_inversi = np.zeros(img.shape, dtype=np.uint8) # Membuat matriks kosong dengan ukuran yang sama dengan gambar.

def inversi_grayscale(nilai):
    # Fungsi untuk melakukan inversi pada gambar dalam skala keabuan.
    for y in range(0, img_height):
        for x in range(0, img_width):
            red = img[y][x][0] # Komponen merah piksel.
            green = img[y][x][1] # Komponen hijau piksel.
            blue = img[y][x][2] # Komponen biru piksel.
            gray = (int(red) + int(green) + int(blue)) / 3 # Menghitung rata-rata komponen untuk menghasilkan skala keabuan.
            gray = nilai - gray # Melakukan inversi pada nilai skala keabuan.
            img_inversi[y][x] = (gray, gray, gray) # Memperbarui nilai piksel dengan skala keabuan yang diinversi.

def inversi_rgb(nilai):
    # Fungsi untuk melakukan inversi pada gambar dalam mode RGB.
    for y in range(0, img_height):
        for x in range(0, img_width):
            red = img[y][x][0] # Komponen merah piksel.
            red = nilai - red # Melakukan inversi pada komponen merah.
            green = img[y][x][1] # Komponen hijau piksel.
            green = nilai - green # Melakukan inversi pada komponen hijau.
            blue = img[y][x][2] # Komponen biru piksel.
            blue = nilai - blue # Melakukan inversi pada komponen biru.
            img_inversi[y][x] = (red, green, blue) # Memperbarui nilai piksel dengan komponen warna yang diinversi.

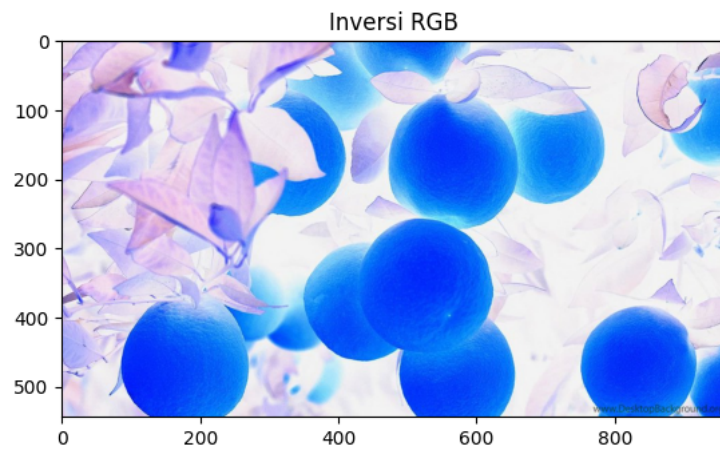
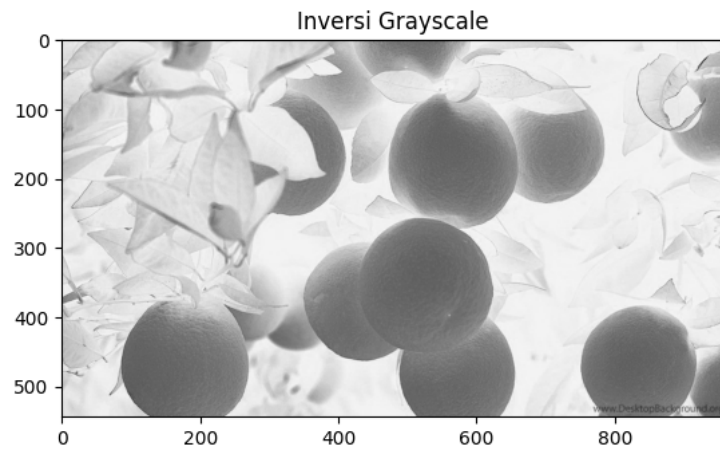
inversi_grayscale(255) # Memanggil fungsi inversi grayscale dengan nilai inversi 255.
plt.imshow(img_inversi) # Menampilkan gambar yang diinversi grayscale.
plt.title("Inversi Grayscale") # Menetapkan judul plot.

```

```
plt.show() # Menampilkan plot gambar inversi grayscale.
```

```
# Inversi RGB  
inversi_rgb(255)
```

```
# Menampilkan gambar hasil inversi  
plt.imshow(img_inversi)  
plt.title("Inversi RGB")  
plt.show()
```



```
# Membuat array kosong untuk gambar logaritmik  
img_log = np.zeros(img.shape, dtype=np.uint8)
```

```
# Fungsi logaritmik untuk setiap piksel  
def log(c):
```

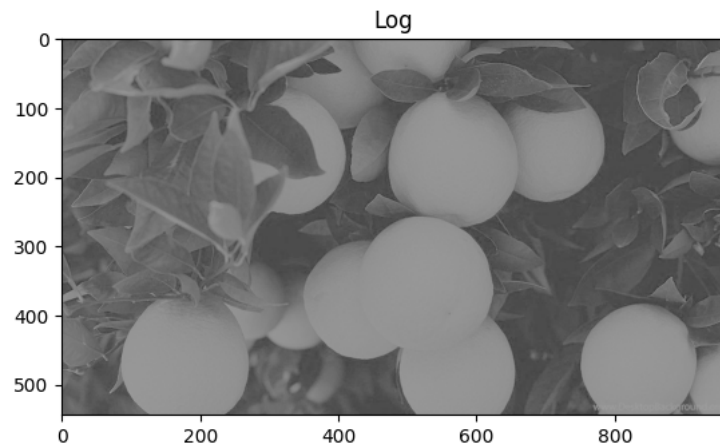
```

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]
        gray = (int(red) + int(green) + int(blue)) / 3
        gray = int(c * np.log(gray + 1))
        if gray > 255:
            gray = 255
        if gray < 0:
            gray = 0
        img_log[y][x] = (gray, gray, gray)

# Mengaplikasikan fungsi logaritmik dengan parameter c = 30
log(30)

# Menampilkan gambar hasil logaritmik
plt.imshow(img_log)
plt.title("Log")
plt.show()

```



```

# Membuat array kosong untuk gambar hasil inversi dan logaritmik
img_inlog = np.zeros(img.shape, dtype=np.uint8)

```

```

# Fungsi inversi dan logaritmik untuk setiap piksel
def inlog(c):
    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]
            gray = (int(red) + int(green) + int(blue)) / 3
            gray = int(c * np.log(255 - gray + 1))
            if gray > 255:
                gray = 255

```

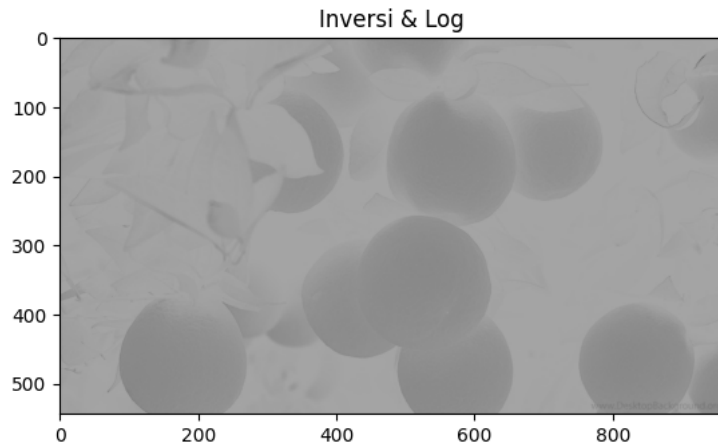
```

if gray < 0:
    gray = 0
img_inlog[y][x] = (gray, gray, gray)

# Mengaplikasikan fungsi inversi dan logaritmik dengan parameter c = 30
inlog(30)

# Menampilkan gambar hasil inversi dan logaritmik
plt.imshow(img_inlog)
plt.title("Inversi & Log")
plt.show()

```



```

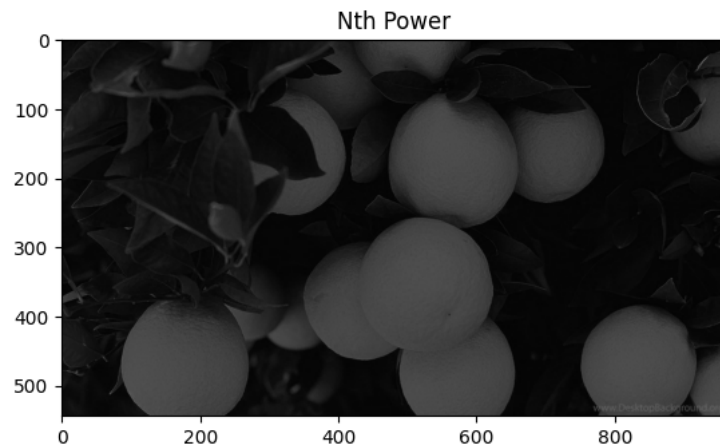
# Membuat array kosong untuk gambar hasil pangkat n
img_nthpower = np.zeros(img.shape, dtype=np.uint8)

# Fungsi pangkat n untuk setiap piksel
def nthpower(c, y):
    thc = c / 100
    thy = y / 100
    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]
            gray = (int(red) + int(green) + int(blue)) / 3
            gray = int(thc * pow(gray, thy))
            if gray > 255:
                gray = 255
            if gray < 0:
                gray = 0
            img_nthpower[y][x] = (gray, gray, gray)

# Menghitung nilai pangkat nth dari 50^100
nthpower(50, 100)

```

```
# Menampilkan gambar hasil pangkat nth
plt.imshow(img_nthpower)
plt.title("Nth Power")
plt.show()
```



```
# Membuat array kosong dengan ukuran yang sama dengan gambar (img)
img_nthrootpower = np.zeros(img.shape, dtype=np.uint8)
```

```
# Fungsi untuk menghitung pangkat akar nth
def nthrootpower(c, y):
    thc = c / 100 # Menghitung persentase konstanta c
    thy = y / 100 # Menghitung persentase konstanta y

    # Melakukan iterasi pada setiap piksel gambar
    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]

            # Menghitung rata-rata warna (grayscale)
            gray = (int(red) + int(green) + int(blue)) / 3

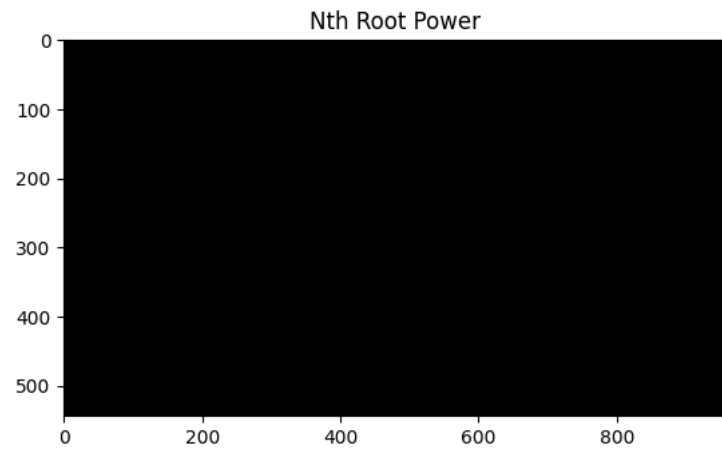
            # Menghitung nilai grayscale setelah dipangkatkan dan diakar pangkat nth
            gray = int(thc * pow(gray, 1./thy))

            # Menyimpan nilai grayscale pada array gambar img_nthpower
            if gray > 255:
                gray = 255
            if gray < 0:
                gray = 0
            img_nthpower[y][x] = (gray, gray, gray)
```

```
# Menghitung pangkat akar nth dengan nilai c=50 dan y=100
```

```
nthrootpower(50, 100)
```

```
# Menampilkan gambar hasil pangkat akar nth  
plt.imshow(img_nthrootpower)  
plt.title("Nth Root Power")  
plt.show()
```



✓ 8s completed at 8:29 PM

