

UAS PENGOLAHAN CITRA

Nama : Rafli Maulana

Kelas: TI 22 B2

NIM: 312210432

1. Kode

```
import numpy as np
import matplotlib.pyplot as plt
import cv2

%matplotlib inline

# untuk membaca gambar gunakan gambar sesuai dengan yg dimiliki
image = cv2.imread('images/monarch.jpg')

# Change color to RGB (from BGR)
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

plt.imshow(image)

# berfungsi untuk Membentuk ulang gambar menjadi susunan piksel 2D dan 3 nilai
warna (RGB)
pixel_vals = image.reshape((-1,3))

# berfungsi untuk mengkonversikan ke tipe float
pixel_vals = np.float32(pixel_vals)

#baris kode di bawah ini menentukan kriteria agar algoritme berhenti berjalan,
#yang akan terjadi adalah 100 iterasi dijalankan atau epsilon (yang merupakan
akurasi yang dibutuhkan)
#menjadi 85%
criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 100, 0.85)

# lalu lakukan k-means clustering dengan jumlah cluster yang ditetapkan sebagai 3
#juga pusat acak pada awalnya dipilih untuk pengelompokan k-means
k = 3

retval, labels, centers = cv2.kmeans(pixel_vals, k, None, criteria, 10,
cv2.KMEANS_RANDOM_CENTERS)
```

```
# mengonversi data menjadi nilai 8-bit
centers = np.uint8(centers)
segmented_data = centers[labels.flatten()]

# membentuk ulang data menjadi dimensi gambar asli
segmented_image = segmented_data.reshape((image.shape))

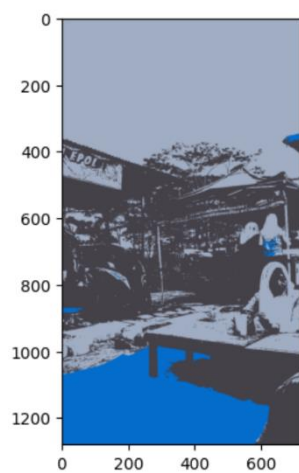
plt.imshow(segmented_image)
```

Gmabar input



Gambar Output

[2]: <matplotlib.image.AxesImage at 0x1d93b469690>



2. KODE

```
import numpy as np

import matplotlib.pyplot as plt

import cv2

%matplotlib inline

# untuk membaca gambar gunakan gambar sesuai dengan yg dimiliki
image = cv2.imread('04072024.jpeg')

# Change color to RGB (from BGR)
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

plt.imshow(image)

# berfungsi untuk Membentuk ulang gambar menjadi susunan piksel 2D dan 3 nilai warna (RGB)
pixel_vals = image.reshape((-1,3))

# berfungsi untuk mengkonversikan ke tipe float
pixel_vals = np.float32(pixel_vals)

#baris kode di bawah ini menentukan kriteria agar algoritme berhenti berjalan,
#yang akan terjadi adalah 100 iterasi dijalankan atau epsilon (yang merupakan akurasi yang
dibutuhkan)

#menjadi 85%
criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 10, 1000)

# lalu lakukan k-means clustering dengan jumlah cluster yang ditetapkan sebagai 3
#juga pusat acak pada awalnya dipilih untuk pengelompokan k-means
k = 5

retval, labels, centers = cv2.kmeans(pixel_vals, k, None, criteria,
25, cv2.KMEANS_RANDOM_CENTERS)

# mengonversi data menjadi nilai 8-bit
centers = np.uint8(centers)

segmented_data = centers[labels.flatten()]

# membentuk ulang data menjadi dimensi gambar asli
segmented_image = segmented_data.reshape((image.shape))

plt.imshow(segmented_image)
```

Output

