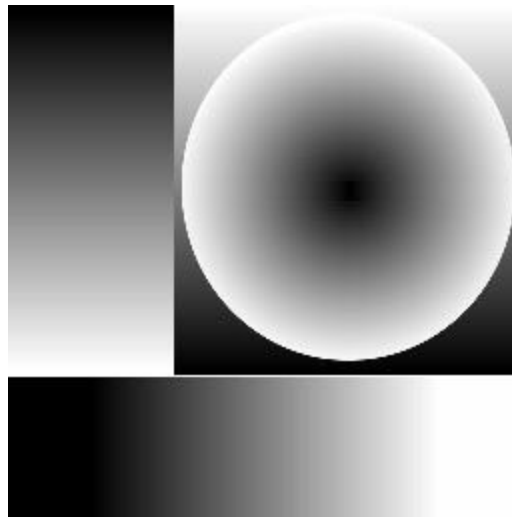


Diberikan gambar sebagai berikut :



Gambar tersebut akan dicoba untuk dikompresi dengan vector quantization. Metode vector quantization yang dilakukan yaitu k-means cluster. Dengan menggunakan package numpy, openCV, dan sklearn.

```
import numpy as np
import os
import cv2
from sklearn.cluster import KMeans
```

Package numpy digunakan untuk konversi array, os digunakan untuk menampilkan size dari gambar, cv2 digunakan untuk membaca dan menampilkan gambar dan yang terakhir sklearn.cluster untuk mengimport fungsi k-means.

```
path_file = "image-soalno-4.bmp"
image = cv2.imread(path_file)

print("Dimensi gambar :", image.shape)
print("Ukuran gambar asli : {} kb".format(os.stat(path_file).st_size /
1024))
```

Masukan gambar ke dalam variabel image. Lihat dimensi awal gambar dan size-nya dalam byte. Selanjutnya, dari gambar yang telah disimpan pada variabel image yang berbentuk array 2 dimensi akan diubah ke bentuk flatten array (array 1 dimensi) agar bisa diolah ke dalam fungsi KMeans.

```
original = np.array(image, dtype=np.float64).reshape((-1, 1))

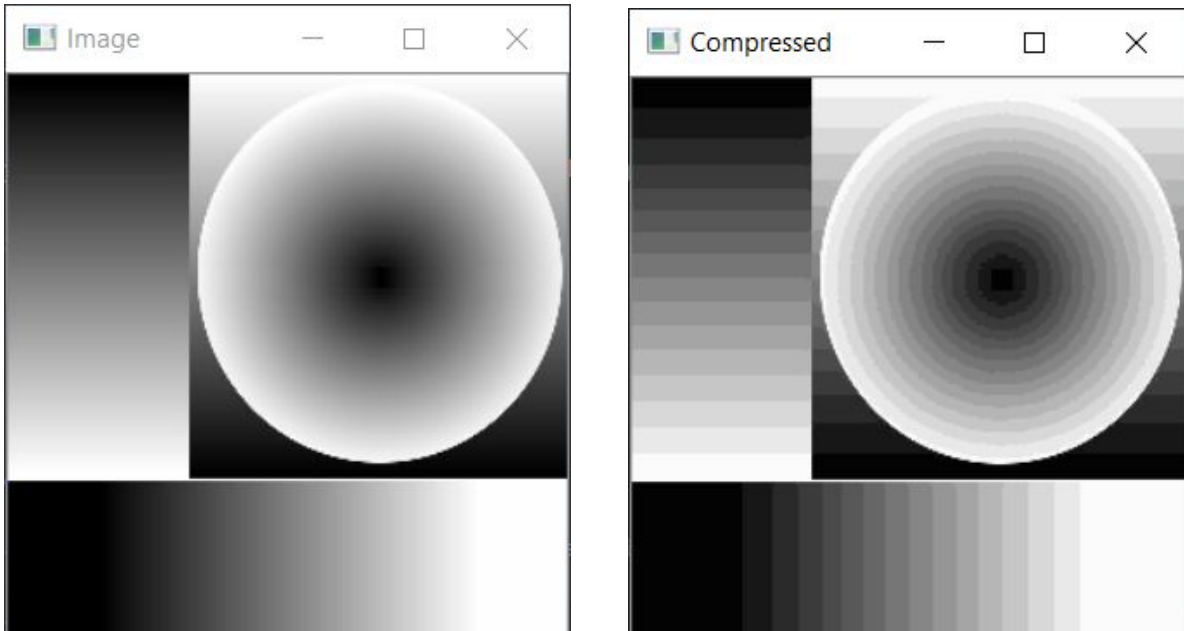
kmeans = KMeans(n_clusters=16, init='random')
kmeans.fit(original)
centers = kmeans.cluster_centers_.squeeze()
labels = kmeans.labels_
```

Klaster yang digunakan berjumlah 16. Klaster ini setara dengan jumlah codebook dalam LBG algorithm dimana nilai-nilai pixel akan dipetakan menjadi 16 nilai. Inisiasi nilai klaster dilakukan secara random. Setelah fungsi KMeans dijalankan, maka akan diperoleh 16 titik center dan array label untuk memetakan nilai.

```
compressed = []
for label in labels:
    compressed.append(centers[label])

compressed = np.asarray(compressed, dtype=np.uint8).reshape(image.shape)
```

Konstruksi ulang gambar yang telah terpetakan ke dalam label-label tersebut menjadi array yang berdimensi sama dengan gambar awal. Berikut hasil gambarnya :



Gambar pertama adalah gambar asli yang diberikan di awal, sedangkan gambar kedua adalah gambar setelah dikompresi dengan vector quantization berjumlah 16 klaster. *Semakin besar nilai kluster, maka akan semakin mendekati ke gambar aslinya begitu pula sebaliknya.* Lalu, hitung nilai psnr menggunakan fungsi pada openCV

```
psnr = cv2.PSNR(image, compressed)
```

Nilai yang diperoleh untuk gambar yang dikompres adalah : 34.84449384832749

Semakin tinggi nilai PSNR akan semakin baik kualitas gambarnya. Untuk penerapan yang lebih baik, bisa dilakukan kompresi jenis lossless dengan algoritma diantaranya Huffman Coding, Lempel-Ziv, dan Run Length Encoding.