

Ahmad Maulvi Alfansuri

G64160081

Soal :

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import cv2
import numpy as np
from pprint import pprint
from sys import *
import matplotlib.pyplot as plt
```

Import setiap module yang dibutuhkan untuk membuat kodingan ini.

```
def calculate(b, g, r):
    grey = r * 0.299 + g * 0.587 + b * 0.114
    return grey
```

Fungsi untuk merubah rgb menjadi gray level

```
def greyscaling_image(image):
    (row, col, chan) = image.shape
    # print (image.shape)
    greyscale = np.zeros((row, col, 1), np.uint8)
    for y in range(row):
        for x in range(col):
            # get pixel
            (b, g, r) = image[y, x]
            # calculate
            grey = calculate(b, g, r)
            # assign
            greyscale.itemset((y, x, 0), grey) # assign
    return greyscale
```

Fungsi untuk merubah image menjadi grayscale

```
def load_image(filename='car.png'):
    image = cv2.imread(filename)
    return image
```

Fungsi meload gambar dan melakukan return image dengan tipe data np.array

```
def extract_image(image):
    pixel = np.array([0 for i in range(256)])
    (row, col, chan) = image.shape
    for y in range(row):
        for x in range(col):
            pix = image[y, x, 0]
            pixel[pix] += 1
    return pixel
```

Fungsi ekstrak image menerima argumen image dan melakukan return frequency pixel yang ada pada gambar

```
def transform(pr):
    rk = np.array([0 for i in range(256)])
    for i in range(256):
        hasil = 0
        for j in range(i + 1):
            hasil += 255 * pr[j]
        rk[i] = round(hasil)
    return rk
```

Fungsi Transform akan mengubah nilai peluang kemunculan pixel menjadi peluang kumulatifnya.

```
def new_data(sk, nk):
    box = {}
    for (idx, i) in enumerate(sk):
        if i in box:
            box[i] += nk[idx]
        elif i not in box:
            box[i] = nk[idx]
    return box
```

Fungsi `new_data` menerima input `sk` yang merupakan array `rk` setelah normalisasi, dan `nk` adalah frekuensi pixel. Setiap `sk` yang sama akan dijumlahkan nilai `nk` nya. Sehingga nilai `box` yang direturn.

```
def normalized(box, n):
    new_box = {}
    for i in box:
        new_box[i] = box[i] * 1.0 / n
    return new_box
```

Fungsi `normalized` menerima array value dan jumlah seluruh data. Setiap `box` akan di assign dengan dibagi dengan jumlah data agar data berada di antara 0 sampai 1

```
def extract_dict(data):
    key = []
    value = []
    for i in data:
        key.append(i)
        value.append(data[i])
    return (key, value)
```

Fungsi Ekstrak data menerima array data dan membagi nilai `key` dan `value` menjadi list. Sehingga dapat diproses dengan lebih mudah.

```
def histogram(data):
    (x, y) = data
    plt.bar(x, y)
    plt.show()
    return 0
```

Fungsi `histogram` akan menampilkan plot berdasarkan data `(x, y)` yang diberikan.

```
def main():
    image = load_image()
    image_greyscale = greyscaling_image(image)
    nk = extract_image(image_greyscale)
    (row, col, chan) = image_greyscale.shape
    n = row * col * 1.0
```

```
pr_rk = nk / n
sk = transform(pr_rk)
new_equal_data = new_data(sk, nk)
normalized_data = normalized(new_equal_data, n)
data = extract_dict(normalized_data)
return_value = histogram(data)
```

Fungsi main adalah fungsi utama algoritma. Langkah langkah program adalah

1. Load image
2. Ubah menjadi grayscale
3. Hitung frekuensi pixel
4. Hitung peluang kemunculan pixel
5. Hitung transformasi pixel
6. Hitung equal data berdasarkan sk, dan nk, sehingga membuat data baru
7. Normalize data yang baru
8. Ekstrak dictionary menjadi dua array yaitu key dan value dalam bentuk array
9. Gambar Histogram dengan matplotlib

```
main()
```

