LAPORAN MATA KULIAH PENGOLAHAN CITRA DIGITAL (PCD)

Dosen Pengampu : Shinta Dwi Angreni, S. Si., M. Kom.

"Implementasi Histogram Equalization, Difference Image dan Image Averaging* menggunakan *Pycharm"

Disusun Oleh :

Isra Septia Cahyani

F551 21 060

B

PROGRAM STUDI S1-TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

FAKULTAS TEKNIK

UNIVERSITAS TADULAKO

2023

A. Histogram Equalization

1. Pertama Mengimpor library OpenCV, numpy, dan pyplot dari matplotlib.

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

```

   - *OpenCV* (*Open Source Computer Vision Library*) merupakan sebuah pustaka perangkat lunak yang digunakan untuk pengolahan citra secara dinamis yang *real-time*. *OpenCV* juga menyediakan berbagai algoritma untuk memproses gambar dan video, termasuk operasi dasar seperti konversi warna, thresholding, filtering, operasi morfologi, deteksi tepi, serta algoritma yang lebih kompleks.
   - *NumPy (Numerical Python)* adalah sebuah library Python yang digunakan untuk melakukan operasi matematika dan manipulasi data numerik secara efisien. NumPy menyediakan objek array multidimensi yang cepat dan efisien serta fungsi matematika yang dapat dioperasikan pada array tersebut.
   - Matplotlib adalah library Python yang digunakan untuk membuat visualisasi data dalam bentuk grafik dan plot. Matplotlib menyediakan berbagai jenis plot seperti line plot, scatter plot, bar plot, histogram, dan masih banyak lagi. Matplotlib juga salah satu library yang paling sering digunakan dalam analisis data dan ilmu data karena mudah digunakan dan memiliki dokumentasi yang lengkap.

2. Lalu, Memuat citra grayscale menggunakan fungsi imread pada OpenCV dengan menggunakan mode grayscale (0)

```
# memuat gambar skala abu-abu
img = cv2.imread('gambar.jpg', 0)

```

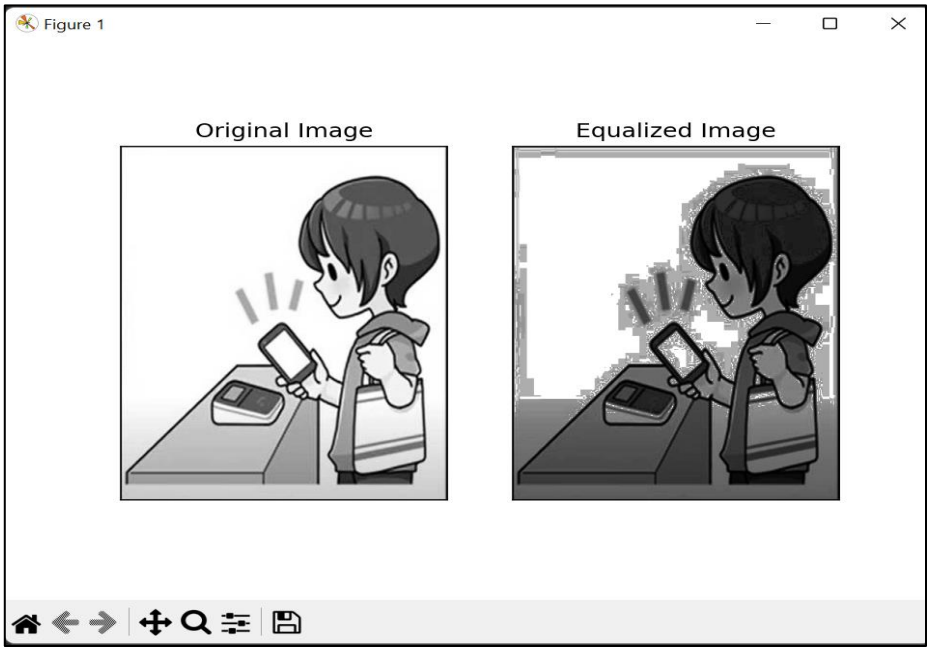
3. Kemudian, Melakukan histogram equalization pada citra menggunakan fungsi equalizeHist pada OpenCV.

```
# melakukan pemerataan histogram
equ = cv2.equalizeHist(img)

```

4. Menampilkan citra asli dan citra yang sudah di-equalize pada sebuah plot menggunakan pyplot dari matplotlib.

```
# menampilkan gambar asli dan disamakan
plt.subplot(121), plt.imshow(img, cmap='gray')
plt.title('Original Image'), plt.xticks([]), plt.yticks([])
plt.subplot(122), plt.imshow(equ, cmap='gray')
plt.title('Equalized Image'), plt.xticks([]), plt.yticks([])
plt.show()
```

Dan di hasilkan gambar dengan dua citra, yaitu citra asli dan citra yang sudah di-equalize. Citra yang sudah di-equalize akan memiliki kontras yang lebih baik dan lebih mudah untuk dianalisis.



B. Difference Image

1. Pertama, Import ibrary *OpenCV* untuk memproses gambar

```
1  import cv2
2
```

   - *OpenCV* (*Open Source Computer Vision Library*) merupakan sebuah pustaka perangkat lunak yang digunakan untuk pengolahan citra secara dinamis yang *real-time*. *OpenCV* juga menyediakan berbagai algoritma untuk memproses gambar dan video, termasuk operasi dasar seperti konversi warna, thresholding, filtering, operasi morfologi, deteksi tepi, serta algoritma yang lebih kompleks.

2. Kemudian membaca dua gambar menggunakan fungsi "*cv2.imread()"* Gambar pertama dan kedua disimpan di variable "*img1*" dan "*img2*".

```
3  # Load gambar pertama dan kedua
4  img1 = cv2.imread('image1.jpg')
5  img2 = cv2.imread('image2.jpg')
6
```

3. lalu mengubah kedua gambar menjadi citra grayscale menggunakan fungsi "*cv2.cvtColor*()". Citra grayscale hanya memiliki satu channel warna, yang berarti informasi warna dihilangkan dan hanya informasi kecerahan yang tetap.

```
# Ubah kedua gambar menjadi grayscale
gray_img1 = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
gray_img2 = cv2.cvtColor(img2, cv2.COLOR_BGR2GRAY)

```

4. selanjutnya, menghitung citra perbedaan antara kedua gambar menggunakan fungsi "*cv2absdiff*()". Fungsi "cv2absdiff()"menghitung selisih absolut antara dua citra grayscale. Citra perbedaan mengandung informasi tentang area yang berbeda antara dua gambar.

```
# Hitung perbedaan pixel antara kedua gambar
diff = cv2.absdiff(gray_img1, gray_img2)

```

5. setelah itu, meng-threshold citra perbedaan untuk menghilangkan noise menggunakan fungsi "*cv2threshold*()". Thresholding digunakan untuk memisahkan piksel-piksel yang penting dalam citra dari yang tidak penting. Fungsi "*cv2threshold*()" mengembalikan dua nilai: nilai ambang dan citra hasil threshold. Nilai ambang dalam kode di atas adalah 30 dan citra hasil threshold disimpan di variable *"thresh"*.

```
# Thresholding pada citra perbedaan untuk menghilangkan noise
thresh = cv2.threshold(diff, 10, 255, cv2.THRESH_BINARY)[1]

```

6. Dan mencari kontur pada citra perbedaan yang telah di-threshold menggunakan fungsi "*cv2.findContours()*". Kontur adalah garis atau batas yang mengelilingi area yang berbeda pada citra. Fungsi "cv2.findContours()" mengembalikan tiga nilai: citra, kontur, dan hierarki kontur. Hanya nilai kontur yang disimpan di variabel "*contours"*.

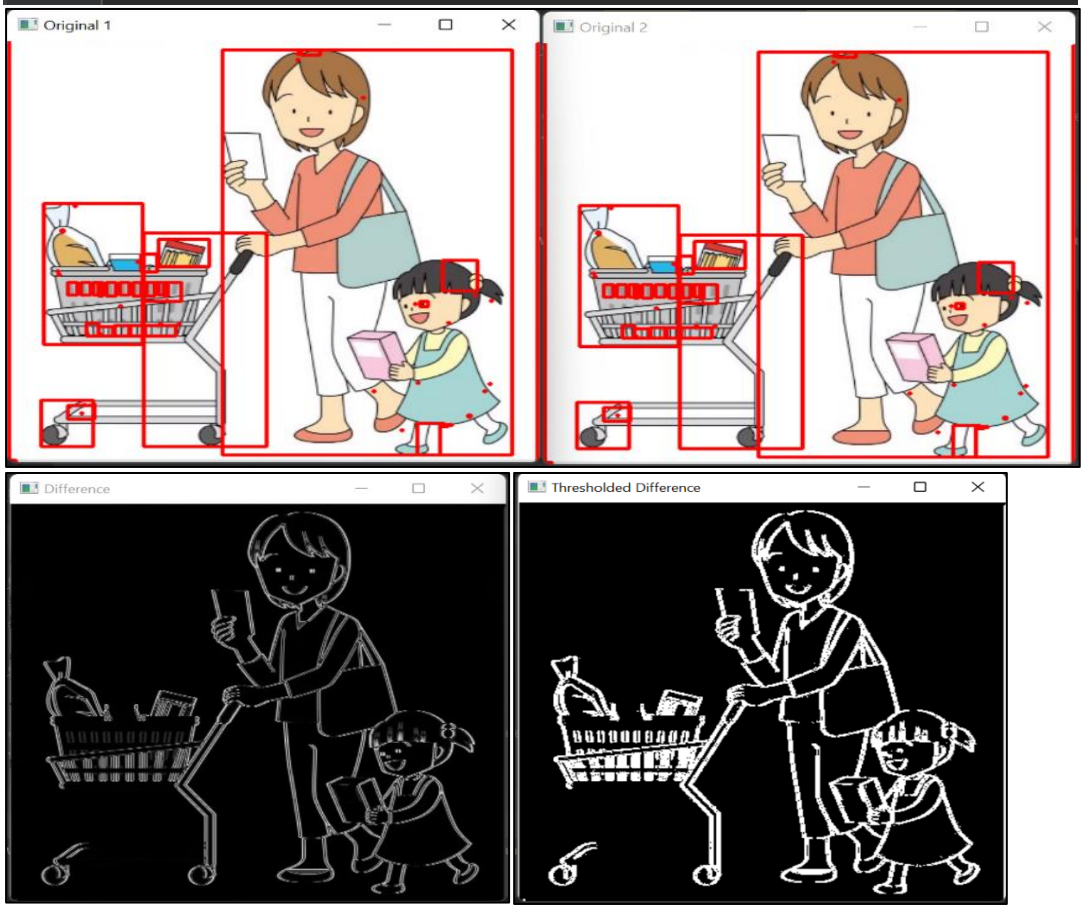
```
# Mencari kontur pada citra perbedaan
contours, hierarchy = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

```

7. Lalu, menggambar kotak pada gambar asli pada area yang berbeda antara dua gambar. Fungsi "*cv2.boundingRect()*" mengembalikan kotak yang melingkupi setiap kontur yang ditemukan. Kotak kemudian digambar pada gambar asli menggunakan fungsi "*cv2.rectangle*()".

```
# Gambar kotak pada kontur yang terdeteksi pada citra asli
for contour in contours:
        (x, y, w, h) = cv2.boundingRect(contour)
        cv2.rectangle(img1, (x, y), (x+w, y+h), (0, 0, 255), 2)
        cv2.rectangle(img2, (x, y), (x+w, y+h), (0, 0, 255), 2)

```

8. Terakhir, menampilkan gambar secara bersamaan dengan dua gambar asli, citra perbedaan, dan citra perbedaan yang telah di-*threshold* dengan kotak yang ditarik pada kontur yang terdeteksi pada gambar asli.

```
# Tampilkan gambar asli dan gambar perbandingan
cv2.imshow("Original 1", img1)
cv2.imshow("Original 2", img2)
cv2.imshow("Difference", diff)
cv2.imshow("Thresholded Difference", thresh)

# Tunggu input dari pengguna
cv2.waitKey(0)
cv2.destroyAllWindows()

```



C. Image Avaraging

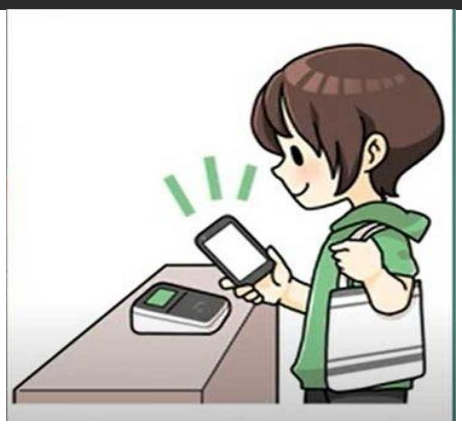
1. Pertama, Mengimpor Numpy dan OpenCV untuk membaca dan menampilkan gambar.

```
import numpy as np
import cv2

```

- *OpenCV* (*Open Source Computer Vision Library*) merupakan sebuah pustaka perangkat lunak yang digunakan untuk pengolahan citra secara dinamis yang *real-time*. *OpenCV* juga menyediakan berbagai algoritma untuk memproses gambar dan video, termasuk operasi dasar seperti konversi warna, thresholding, filtering, operasi morfologi, deteksi tepi, serta algoritma yang lebih kompleks.
- *NumPy (Numerical Python)* adalah sebuah library Python yang digunakan untuk melakukan operasi matematika dan manipulasi data numerik secara efisien. NumPy menyediakan objek array multidimensi yang cepat dan efisien serta fungsi matematika yang dapat dioperasikan pada array tersebut.

9. Lalu membaca gambar dengan menggunakan fungsi "*cv2.imread"*

```
# membaca gambar
img = cv2.imread('gambar.jpg')
```



10. Kemudian mengubah atau menentukan ukuran karnel "ksize" yang akan digunakan untuk filter rata-rata. ukuran karnel berupa bilangan ganjil untuk memastikan karnel memiliki titik tengah.

```
# menentukan ukuran kernel
ksize = 5
```

11. Setelah itu, membuat karnel filter rata-rata dengan menggunakan fungsi "np.ones" untuk membuat matriks berisi 1 dan kemudian membaginya dengan jumlah elemen karnel yaitu "ksize**2" agar rata-rata dari semua elemen karnel menjadi 1.

```
# membuat kernel filter rata-rata
kernel = np.ones((ksize, ksize), np.float32) / (ksize**2)
```

12. Selanjutnya, melakukan konvolusi pada gambar asli menggunakan fungsi “cv2filter2D” dan menyimpan hasilnya dalam variabel

```
# melakukan konvolusi pada gambar menggunakan kernel filter rata-rata
filtered_img = cv2.filter2D(img, -1, kernel)
```

13. Terakhir, menampilkan gambar asli dan hasil filter menggunakan fungsi “cv2.imshow” dan “cv2.destroyAllWindows”

```
# menampilkan gambar asli dan hasil filter
cv2.imshow('Original Image', img)
cv2.imshow('Filtered Image', filtered_img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```