LAPORAN

PENGOLAHAN CITRA DIGITAL (PCD)

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

"Implementasi Difference Image dan Image Averaging menggunakan Pycharm"

Oleh :

Tiara Juli Arsita

F55121053

PROGRAM STUDI S1-TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

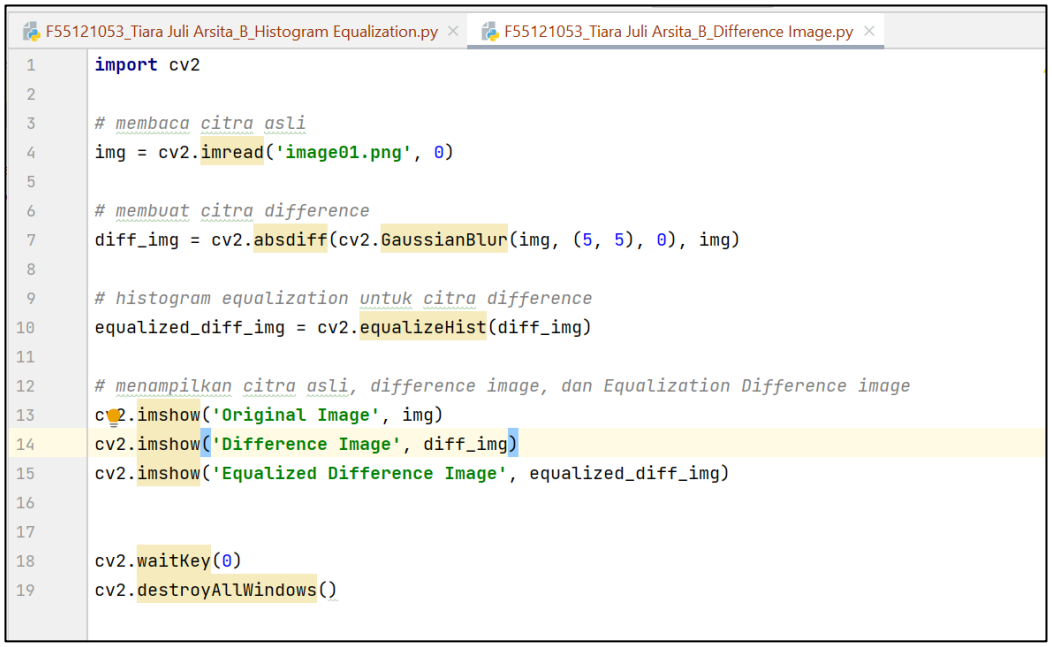
FAKULTAS TEKNIK

UNIVERSITAS TADULAKO

2023

A. *Difference Image*

1. kode program

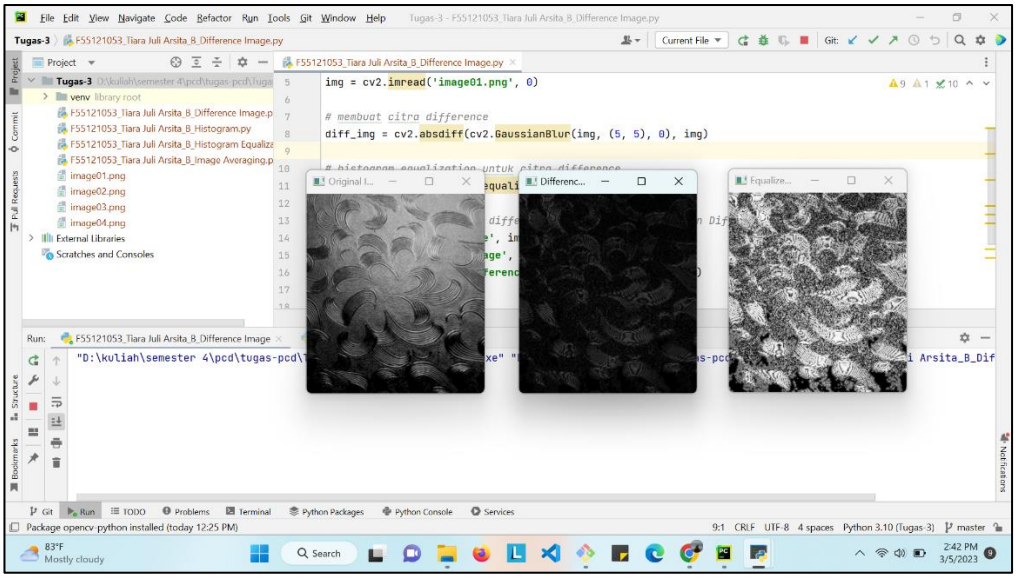


Kode tersebut adalah program Python yang menggunakan OpenCV untuk membaca gambar dari file, membuat citra difference, dan menampilkan citra asli, difference image, dan histogram equalization difference image.

1. import cv2: Memuat pustaka OpenCV ke dalam program.
2. img = cv2.imread('image01.png', 0): Membaca gambar dari file 'image01.png' dengan mode grayscale (0) dan menyimpannya ke variabel img.
3. diff_img = cv2.absdiff(cv2.GaussianBlur(img, (5, 5), 0), img): Membuat citra difference dari gambar yang telah dibaca dengan cara melakukan operasi pengurangan antara gambar asli dan gambar yang telah di-blur menggunakan filter Gaussian.
4. equalized_diff_img = cv2.equalizeHist(diff_img): Mengaplikasikan histogram equalization pada citra difference yang telah dibuat sebelumnya.
5. cv2.imshow('Original Image', img): Menampilkan gambar asli dengan judul 'Original Image'.
6. cv2.imshow('Difference Image', diff_img): Menampilkan citra difference dengan judul 'Difference Image'.

7. cv2.imshow('Equalized Difference Image', equalized_diff_img): Menampilkan histogram equalization difference image dengan judul 'Equalized Difference Image'.
8. cv2.waitKey(0): Menunggu pengguna menekan tombol keyboard. Jika tombol apa pun ditekan, fungsi ini akan mengembalikan nilai yang sesuai dengan tombol yang ditekan.
9. cv2.destroyAllWindows(): Menutup semua jendela tampilan gambar.

2. Hasil Run



B. *Image averaging*

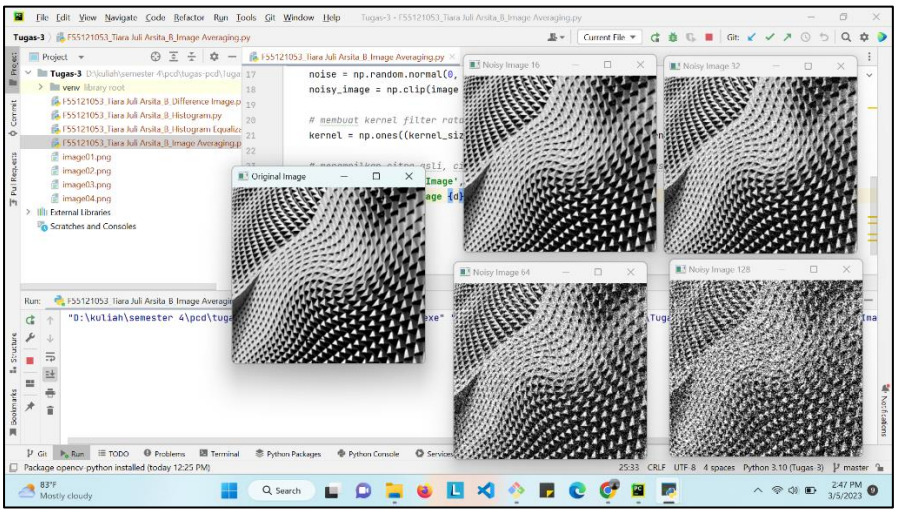
1. kode program

```python
import cv2
import numpy as np
# menginisialisasi citra asli
image = cv2.imread('image02.png', 0)

# mengatur ukuran kernel filter
kernel_size = 3
# mengatur variasi deviasi standard pada noise Gaussian
deviations = [16, 32, 64, 128]

# melakukan filter rata-rata pada masing-masing citra bernoise
for d in deviations:

    # menambahkan noise Gaussian pada citra asli
    noise = np.random.normal(0, d, size=image.shape)
    noisy_image = np.clip(image + noise, 0, 255).astype(np.uint8)
    # membuat kernel filter rata-rata
    kernel = np.ones((kernel_size, kernel_size), np.float32) / (kernel_size**2)
    # menampilkan citra asli, citra yang diberi noise, dan citra hasil filter
    cv2.imshow('Original Image', image)
    cv2.imshow(f'Noisy Image {d}', noisy_image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Kode tersebut adalah program Python yang menggunakan OpenCV untuk membaca gambar dari file, menambahkan noise Gaussian pada gambar, dan melakukan filter rata-rata pada gambar yang telah ditambahkan noise. Setiap citra yang diberi noise dengan variasi deviasi yang berbeda akan ditampilkan pada jendela tampilan.

1. import cv2: Memuat pustaka OpenCV ke dalam program.
2. import numpy as np: Memuat pustaka numpy ke dalam program dengan alias np.
3. image = cv2.imread('image02.png', 0): Membaca gambar dari file 'image02.png' dengan mode grayscale (0) dan menyimpannya ke variabel image.
4. kernel_size = 3: Mengatur ukuran kernel filter rata-rata yang akan digunakan.
5. deviations = [16, 32, 64, 128]: Mengatur variasi deviasi standard pada noise Gaussian yang akan ditambahkan pada citra asli.
6. for d in deviations:: Memulai loop untuk melakukan filter rata-rata pada masing-masing citra bernoise.
7. noise = np.random.normal(0, d, size=image.shape): Menambahkan noise Gaussian pada citra asli dengan variasi deviasi sebesar d.
8. noisy_image = np.clip(image + noise, 0, 255).astype(np.uint8): Membuat citra baru dengan menambahkan noise Gaussian pada citra asli dan mengatur nilai piksel yang dihasilkan agar berada dalam range 0-255.
9. kernel = np.ones((kernel_size, kernel_size), np.float32) / (kernel_size**2): Membuat kernel filter rata-rata dengan ukuran kernel_size x kernel_size.
10. cv2.imshow('Original Image', image): Menampilkan gambar asli dengan judul 'Original Image'.
11. cv2.imshow(f'Noisy Image {d}', noisy_image): Menampilkan citra yang diberi noise dengan variasi deviasi sebesar d dengan judul 'Noisy Image {d}'.
12. cv2.waitKey(0): Menunggu pengguna menekan tombol keyboard. Jika tombol apa pun ditekan, fungsi ini akan mengembalikan nilai yang sesuai dengan tombol yang ditekan.
13. cv2.destroyAllWindows(): Menutup semua jendela tampilan gambar.

2. Hasil run



C. Histogram

1. kode program

F55121053_Tiara Juli Arsita_B_Histogram.py

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

# membaca citra
image = cv2.imread('image03.png', 0)

# membuat histogram
hist, bins = np.histogram(image.flatten(), 256, [0, 256])

# menampilkan histogram sebelum perbaikan
plt.hist(image.flatten(), 256, [0, 256])
plt.xlim([0, 256])
plt.show()
# mencari nilai intensitas piksel dengan frekuensi terbanyak
max_intensity = np.argmax(hist)

# membuat lookup table untuk menggeser nilai intensitas piksel
lut = np.zeros((256,), dtype=np.uint8)
for i in range(256):
    lut[i] = np.uint8(np.clip((i - max_intensity) * 1.5 + max_intensity, 0, 255))

# mengaplikasikan lookup table ke citra
rst = cv2.LUT(image, lut)

# membuat histogram setelah perbaikan
hist_result, bins_result = np.histogram(rst.flatten(), 256, [0, 256])

# menampilkan histogram setelah perbaikan
plt.hist(rst.flatten(), 256, [0, 256])
plt.xlim([0, 256])
plt.show()
# menampilkan citra asli dan hasil perbaikan
cv2.imshow('Original image', image)
cv2.imshow('Histogram image', rst)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

Kode tersebut merupakan implementasi dari teknik histogram equalization pada citra grayscale menggunakan OpenCV dan NumPy. Histogram equalization adalah sebuah teknik yang digunakan untuk meningkatkan kontras pada citra dengan cara menyeimbangkan distribusi intensitas piksel pada citra.

Pada kode tersebut, citra grayscale dibaca menggunakan fungsi cv2.imread dengan mode bacaan 0. Kemudian, histogram citra dihitung menggunakan fungsi np.histogram, dan ditampilkan pada plot menggunakan plt.hist.

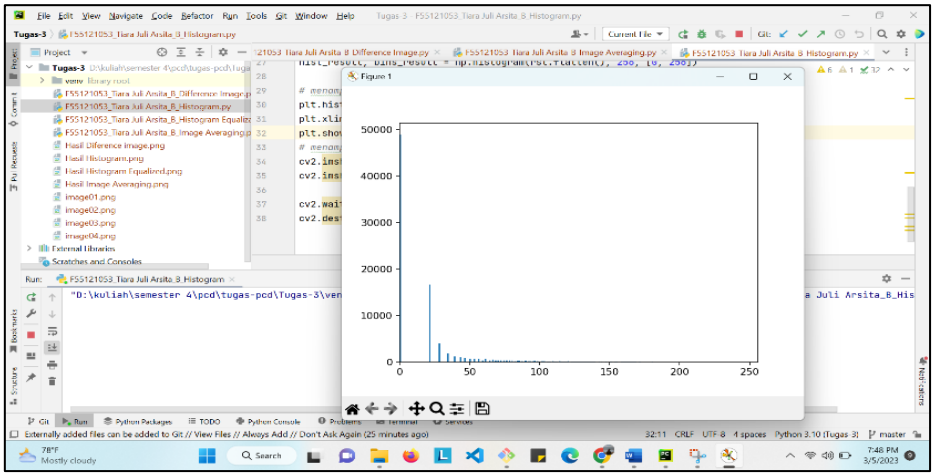
Selanjutnya, nilai intensitas piksel dengan frekuensi terbanyak dicari menggunakan np.argmax(hist). Lookup table (LUT) dibuat dengan menggeser intensitas piksel dengan memperbesar rentang intensitas piksel yang lebih kecil daripada rentang intensitas piksel yang lebih besar, sehingga kontras pada citra meningkat.

Setelah LUT dibuat, citra grayscale asli diperoleh dari fungsi cv2.LUT yang mengaplikasikan LUT pada citra. Histogram citra setelah perbaikan dihitung dan ditampilkan pada plot.

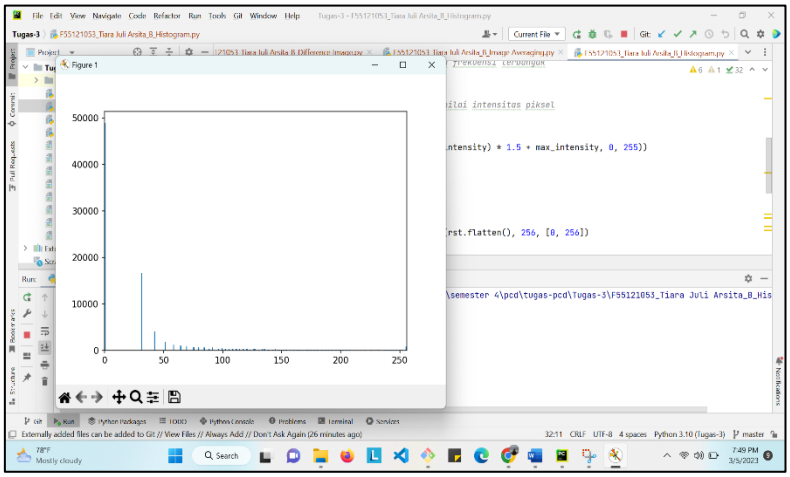
Terakhir, citra asli dan citra hasil perbaikan ditampilkan menggunakan cv2.imshow. Fungsi cv2.waitKey menunggu hingga pengguna menekan tombol pada keyboard, dan kemudian citra ditutup menggunakan cv2.destroyAllWindows.

2. Grafik Histogram

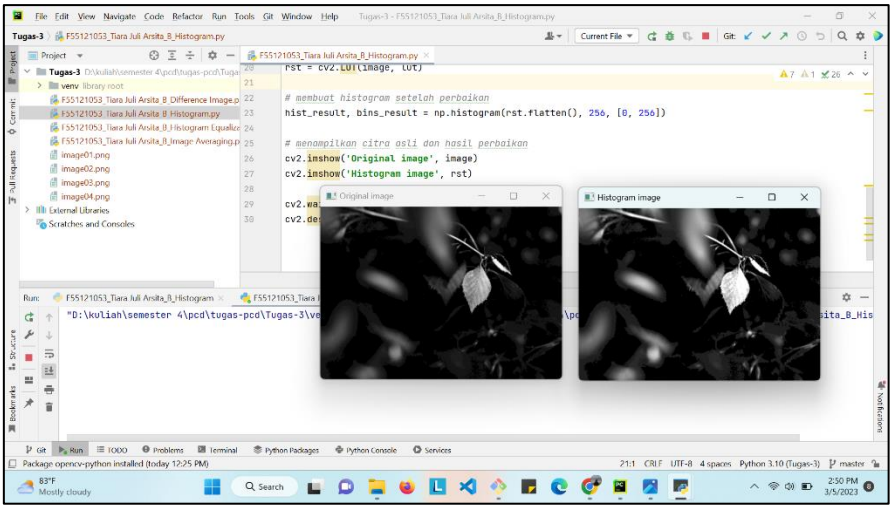
- sebelum perbaikan



- setelah perbaikan



3. Hasil run program



D. Histogram Equalized

1. kode program

```python
import cv2
import numpy as np
from matplotlib import pyplot as plt
# membaca citra
image = cv2.imread('image04.png', 0)

# membuat histogram
hist, bins = np.histogram(image.flatten(), 256, [0, 256])

# menampilkan histogram sebelum perbaikan
plt.hist(image.flatten(), 256, [0, 256])
plt.xlim([0, 256])
plt.show()

# melakukan ekualisasi histogram
equalized_img = cv2.equalizeHist(image)

# membuat histogram setelah perbaikan
hist_result, bins_result = np.histogram(equalized_img.flatten(), 256, [0, 256])
```

```
20
21  # menampilkan histogram setelah perbaikan
22  plt.hist(equalized_img.flatten(), 256, [0, 256])
23  plt.xlim([0, 256])
24  plt.show()
25
26  # menampilkan citra asli dan hasil perbaikan
27  cv2.imshow('Original Image', image)
28  cv2.imshow('Equalized Image', equalized_img)
29
30  cv2.waitKey(0)
31  cv2.destroyAllWindows()
```

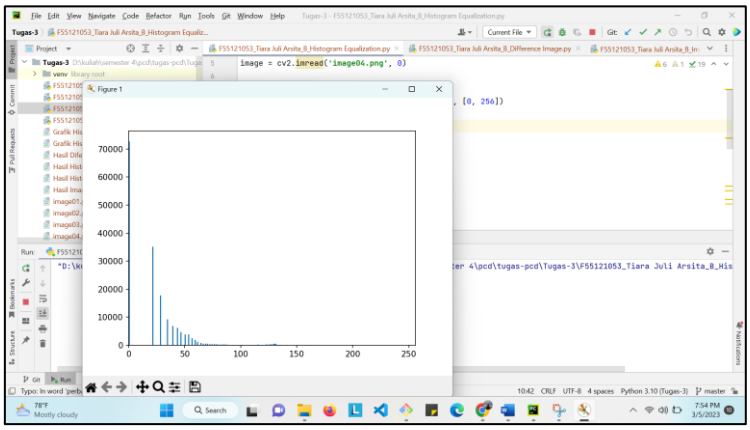
Kode tersebut merupakan contoh implementasi dari histogram equalization menggunakan library OpenCV pada bahasa pemrograman Python. Berikut penjelasan setiap bagian kode:

1. import cv2: mengimport library OpenCV untuk memproses gambar.
2. import numpy as np: mengimport library numpy untuk kebutuhan pengolahan array.
3. from matplotlib import pyplot as plt: mengimport pyplot dari matplotlib untuk menampilkan histogram.
4. image = cv2.imread('image04.png', 0): membaca citra dengan nama 'image04.png' dalam mode grayscale (0).
5. hist, bins = np.histogram(image.flatten(), 256, [0, 256]): membuat histogram dari citra dengan 256 bins, yaitu rentang nilai piksel 0-255.
6. plt.hist(image.flatten(), 256, [0, 256]): menampilkan histogram citra sebelum dilakukan perbaikan.
7. equalized_img = cv2.equalizeHist(image): melakukan ekualisasi histogram pada citra menggunakan fungsi cv2.equalizeHist().
8. hist_result, bins_result = np.histogram(equalized_img.flatten(), 256, [0, 256]): membuat histogram dari citra setelah dilakukan perbaikan.
9. plt.hist(equalized_img.flatten(), 256, [0, 256]): menampilkan histogram citra setelah dilakukan perbaikan.
10. cv2.imshow('Original Image', image): menampilkan citra asli dengan judul 'Original Image'.
11. cv2.imshow('Equalized Image', equalized_img): menampilkan citra setelah dilakukan perbaikan dengan judul 'Equalized Image'.

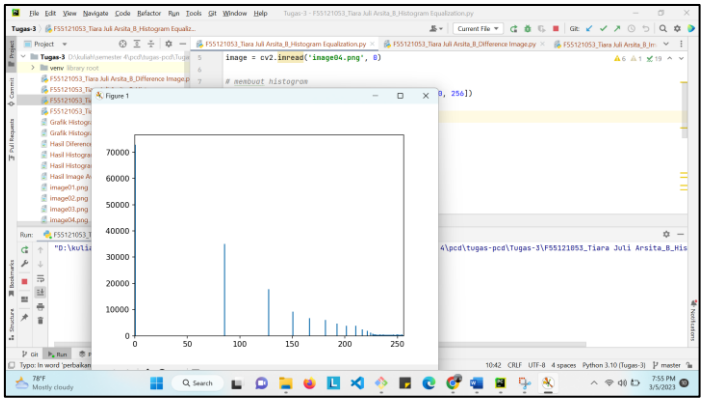
12. cv2.waitKey(0): menunggu pengguna menekan tombol keyboard untuk menutup citra.
13. cv2.destroyAllWindows(): menutup semua jendela citra yang terbuka.

2. Grafik equalization Histogram

- Sebelum perbaikan



- Setelah perbaikan



3. Hasil Run