**PRAKTIKUM 9 EDGE DETECTION**

Nama : Muhamad Arif Maulana

NIM : 1207070069

Link : <https://github.com/velmores/1207070069_Muhamad-Arif-Maulana_Prak9.git>

**Latihan 1 Filter Sobel**

Source Code

import cv2

import numpy as np

from matplotlib import pyplot as plt

from skimage import data

image = cv2.imread("foto/arif.jpeg",0)

image\_sobelx = cv2.Sobel(image, cv2.CV\_8U, 1, 0, ksize = 5)

image\_sobely = cv2.Sobel(image, cv2.CV\_8U, 0, 1, ksize = 5)

image\_sobel = image\_sobelx + image\_sobely

fig, axes = plt.subplots (4, 2, figsize = (20, 20))

ax = axes.ravel ()

ax[0].imshow(image, cmap = 'gray')

ax[0].set\_title ("Citra Input")

ax[1].hist (image.ravel(), bins = 256)

ax[1].set\_title ("Histogram Citra Input")

ax[2].imshow(image\_sobelx, cmap = 'gray')

ax[2].set\_title ("Citra Input")

ax[3].hist (image\_sobelx.ravel(), bins = 256)

ax[3].set\_title ("Histogram Citra Output")

ax[4].imshow(image\_sobely, cmap = 'gray')

ax[4].set\_title ("Citra Input")

ax[5].hist (image\_sobely.ravel(), bins = 256)

ax[5].set\_title ("Histogram Citra Output")

ax[6].imshow(image\_sobel, cmap = 'gray')

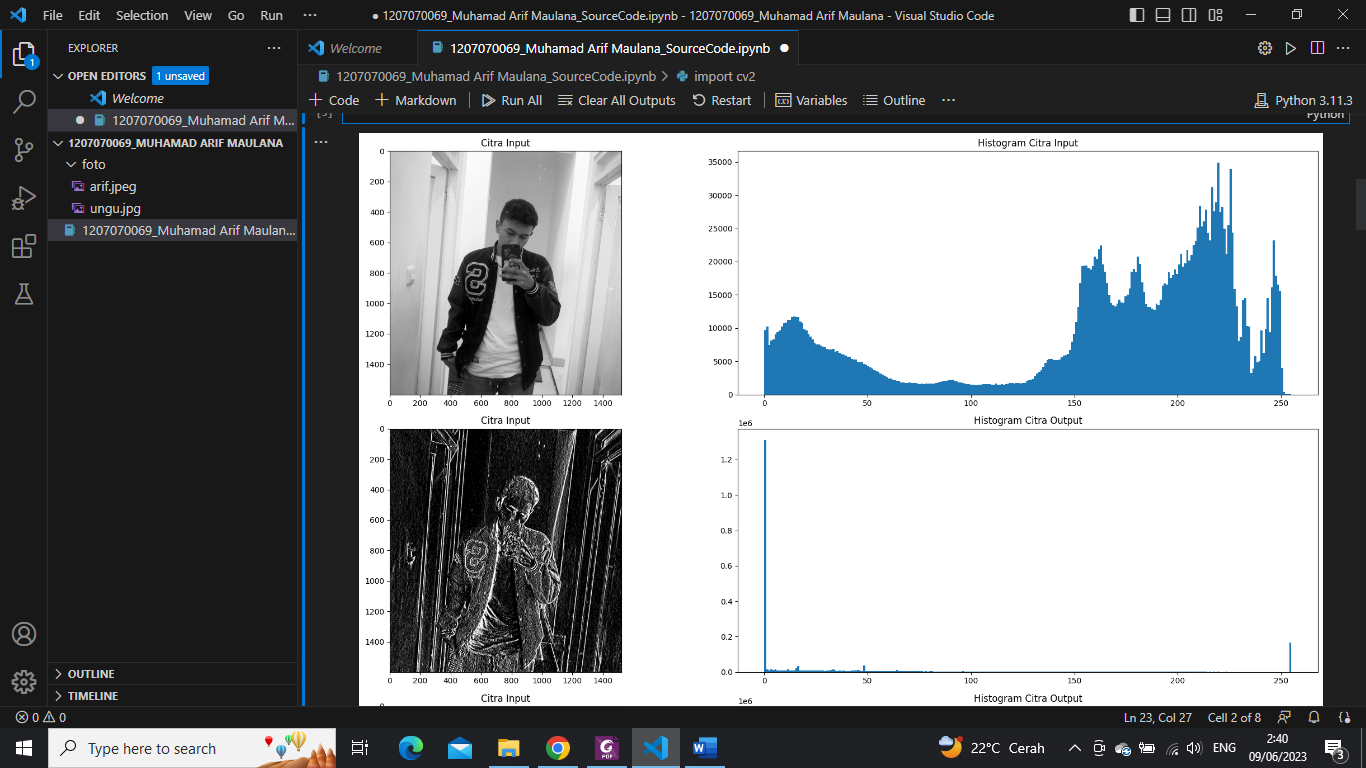
ax[6].set\_title ("Citra Input")

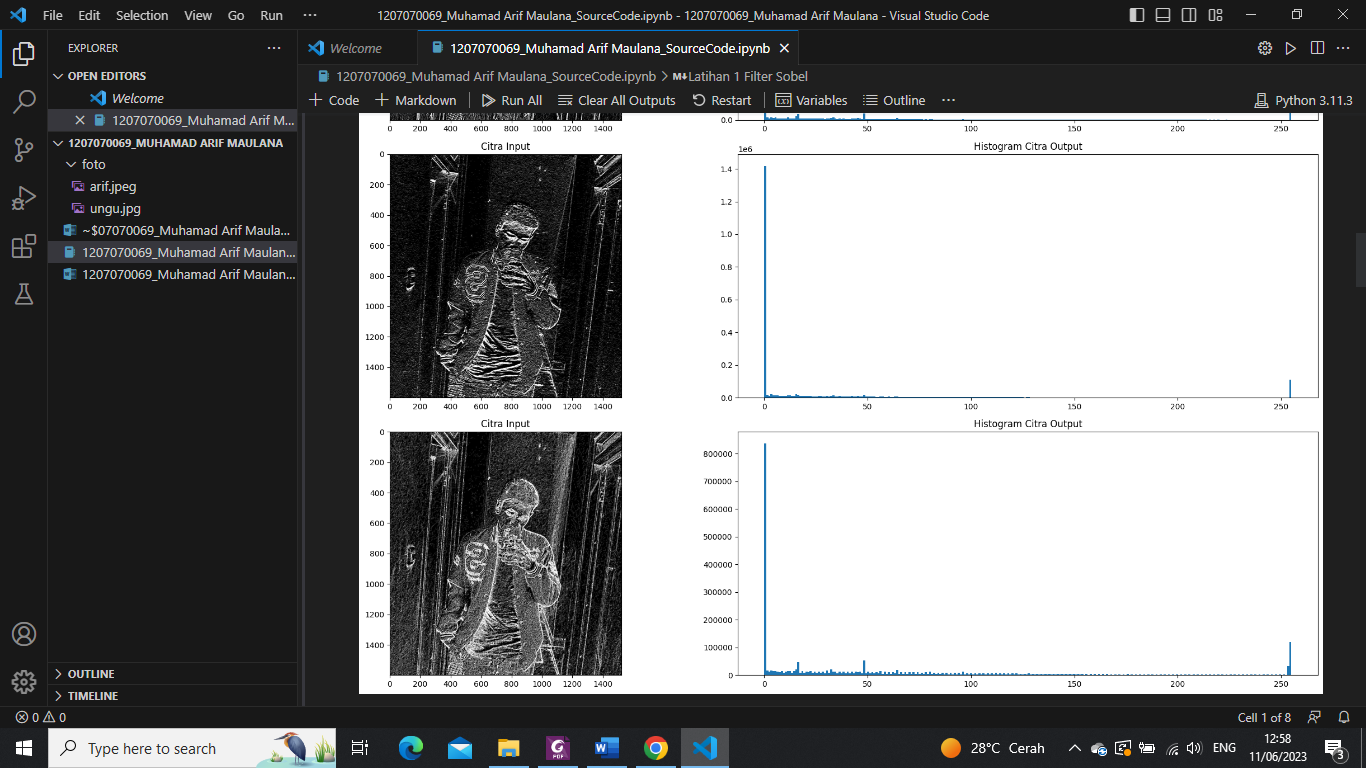
ax[7].hist (image\_sobel.ravel(), bins = 256)

ax[7].set\_title ("Histogram Citra Output")

fig.tight\_layout()

Hasil





**Latihan 2 Filter Prewitt**

Source Code

import cv2

import numpy as np

from matplotlib import pyplot as plt

from skimage import data

image = cv2.imread("foto/arif.jpeg",0)

kernelx = np.array([[1,1,1], [0,0,0], [-1,-1,-1]])

kernely = np.array([[-1,0,1], [-1,0,1], [-1,0,1]])

image\_prewittx = cv2.filter2D(image, -1, kernelx)

image\_prewitty = cv2.filter2D(image, -1, kernely)

image\_prewitt = image\_prewittx + image\_prewitty

fig, axes = plt.subplots (4, 2, figsize = (20, 20))

ax = axes.ravel ()

ax[0].imshow(image, cmap = 'gray')

ax[0].set\_title ("Citra Input")

ax[1].hist (image.ravel(), bins = 256)

ax[1].set\_title ("Histogram Citra Input")

ax[2].imshow(image\_prewittx, cmap = 'gray')

ax[2].set\_title ("Citra Output Prewitt X")

ax[3].hist (image\_prewittx.ravel(), bins = 256)

ax[3].set\_title ("Histogram Citra Output Prewitt X")

ax[4].imshow(image\_prewitty, cmap = 'gray')

ax[4].set\_title ("Citra Output Prewitt Y")

ax[5].hist (image\_sobely.ravel(), bins = 256)

ax[5].set\_title ("Histogram Citra Output Prewitt Y")

ax[6].imshow(image\_prewitt, cmap = 'gray')

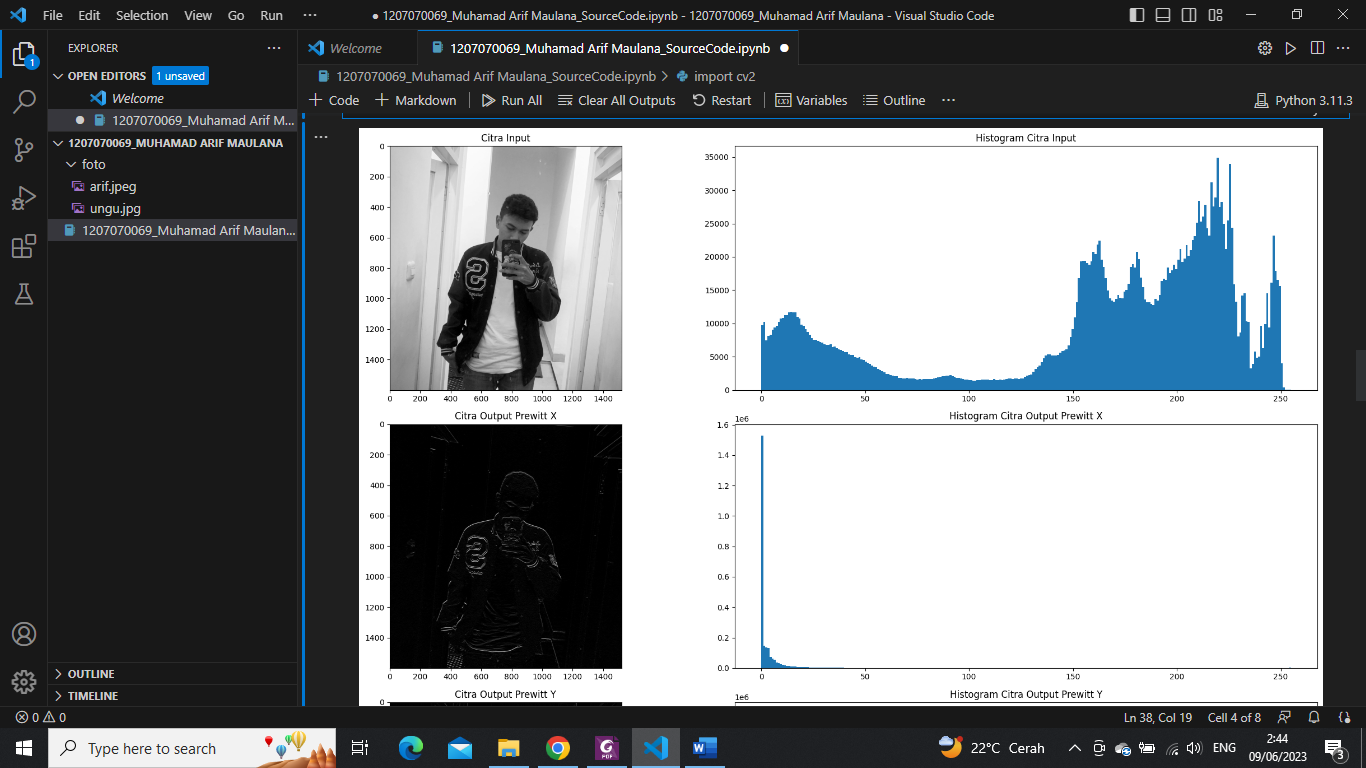
ax[6].set\_title ("Citra Output Prewitt")

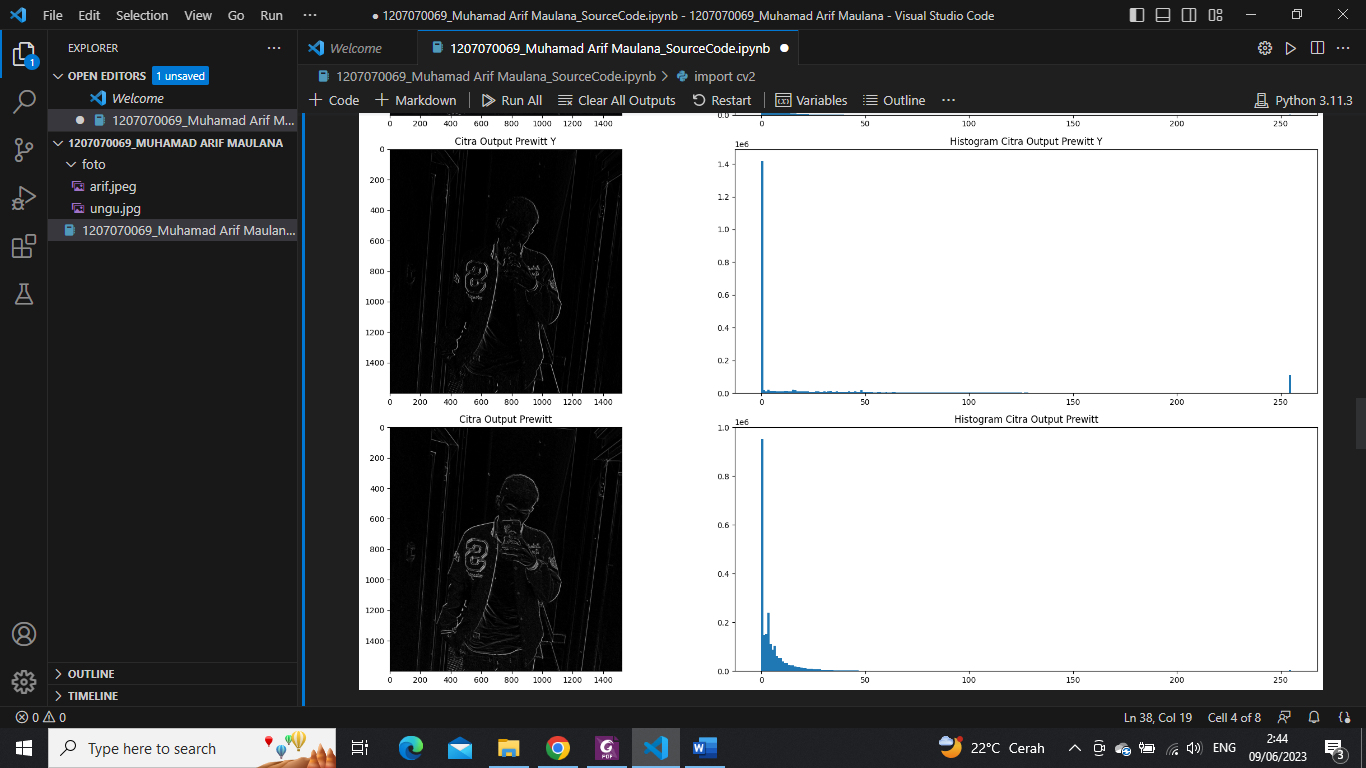
ax[7].hist (image\_prewitt.ravel(), bins = 256)

ax[7].set\_title ("Histogram Citra Output Prewitt")

fig.tight\_layout()

Hasil





**Latihan 3 Filter Canny**

Source Code

import cv2

from matplotlib import pyplot as plt

image = cv2.imread('foto/ungu.jpg',0)

img\_canny = cv2.Canny(image,100,200)

fig, axes = plt.subplots(2, 2, figsize=(6, 6))

ax = axes.ravel()

ax[0].imshow(image, cmap ='gray')

ax[0].set\_title("Citra Input")

ax[1].hist(image.ravel(), bins = 256)

ax[1].set\_title("Histogram Citra Input")

ax[2].imshow(img\_canny, cmap ='gray')

ax[2].set\_title("Citra Output")

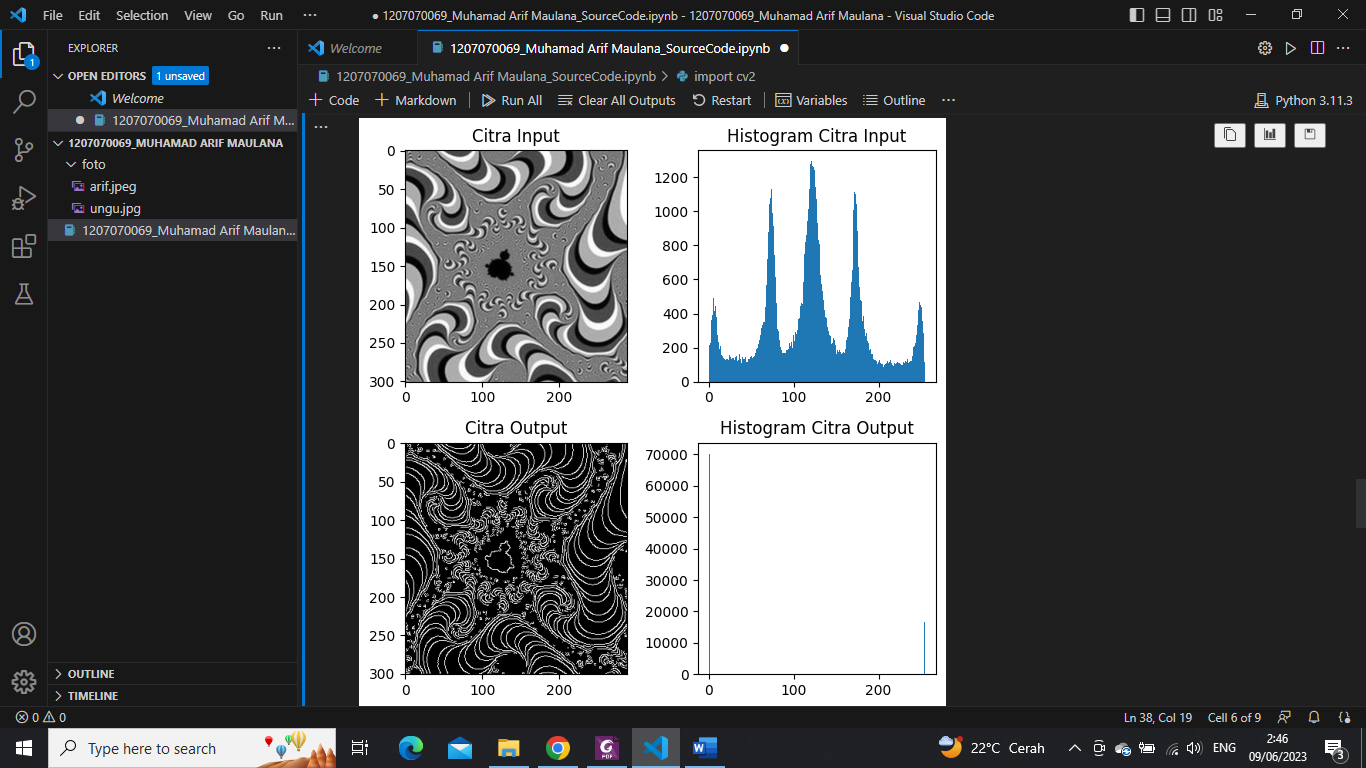
ax[3].hist(img\_canny.ravel(), bins = 256)

ax[3].set\_title("Histogram Citra Output")

fig.tight\_layout()

plt.show()

Hasil



**Latihan 4 Filter Laplacian**

Source Code

import cv2

from matplotlib import pyplot as plt

image0 = cv2.imread('foto/ungu.jpg')

gray = cv2.cvtColor(image0, cv2.COLOR\_BGR2GRAY)

image = cv2.GaussianBlur(gray,(3,3),0)

laplacian = cv2.Laplacian(image,cv2.CV\_64F)

plt.subplot(1,2,1)

plt.imshow(image, cmap='gray')

plt.title('Original')

plt.xticks([])

plt.yticks([])

plt.subplot(1,2,2)

plt.imshow(laplacian, cmap='gray')

plt.title('Laplacian')

plt.xticks([])

plt.yticks([])

plt.show()

img = cv2.imread('foto/ungu.jpg',0)

blur = cv2.GaussianBlur(img,(3,3),0)

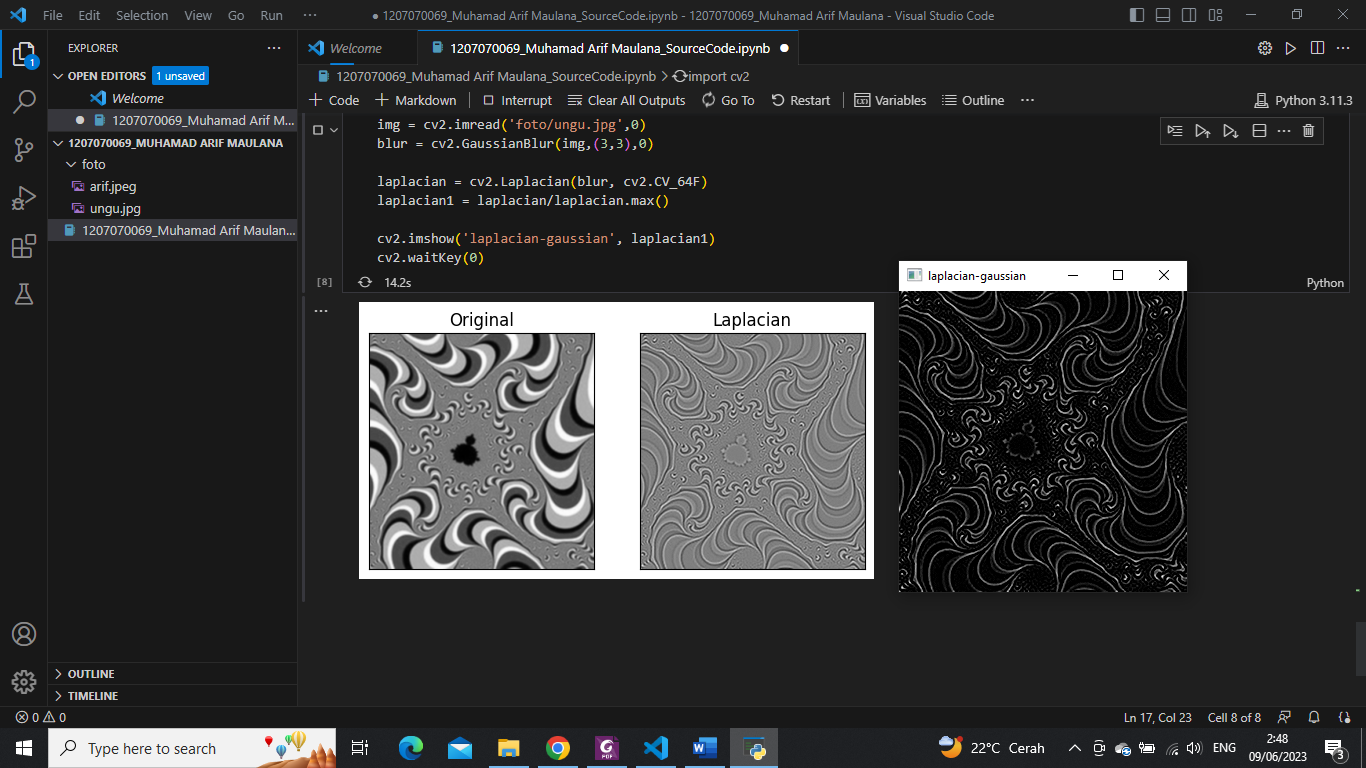
laplacian = cv2.Laplacian(blur, cv2.CV\_64F)

laplacian1 = laplacian/laplacian.max()

cv2.imshow('laplacian-gaussian', laplacian1)

cv2.waitKey(0)

Hasil



**Hasil Pengamatan**

Pada praktikum 9 ini melakukan percobaan Filter Sobel, Filter Prewitt, Filter Canny dan Filter Laplacian. Beberapa filter ini sama digunakan untuk mendeteksi tepi atau perubahan intensitas yang signifikan dalam citra.

1. Percobaan pertama ini melakukan deteksi tepi dengan menggunakan Filter Sobel, filter ini digunakan untuk mencari tepi horizontal dan vertikal. pada praktikum ini terdapat noise tambahan yang menghasilkan gambar output tidak terlalu baik.
2. Pada percobaan kedua ini melakukan deteksi tepi dengan menggunakan Filter Prewitt yang dimana filter ini sama dengan filter sobel, namun pada filter ini tidak memengaruhi noise dan hasil gambar percobaan kedua ini cukup baik.
3. Pada percobaan ketiga ini melakukan deteksi tepi dengan menggunakan Filter Canny. Pada filter ini mampu mendeteksi tepi gambar dengan baik yang dimana ketebalan dan jarak antar tepi sesuai dengan gambar asli dan juga pada gambar tidak tedapat noise seperti pada percobaan sebelumnya.
4. Pada percobaan keempat ini melakukan deteksi tepi dengan menggunakan Filter Laplacian. Pada filter ini hanya menggunakan satu kernel, kemudian filter ini menghitung turunan orde kedua dalam satu lintasan. Percobaan keempat ini menghasilkan gambar yang cukup baik, untuk source code pertama menghasilkan gambar yang lebih halus dan tidak terlalu mendeteksi tepi gambar sedangkan pada source code kedua menghasilkan gambar dan deteksi tepi yang tajam.