

PEMROGRAMAN PENGELOLAAN CITRA

DENGAN PYTHON & MATLAB

Nama	:	Rasheed Mirza Maulana
NPM	:	1942410

1. CITRA BINER

Merubah / transformasi citra grayscale ke dalam bentuk citra biner

$$f(x, y)' = \begin{cases} a_1, & f(x, y) < T \\ a_2, & f(x, y) \geq T \end{cases}$$

Python

```
from PIL import Image

# Create Biner Image
def citra_biner(nilai_ambang):
    citra = Image.open('D:/IMAGE PROCESSING/MATERIAL/lena_grayscale.jpg')
    pixel_citra = citra.load()

    x = citra.size[0]
    y = citra.size[1]

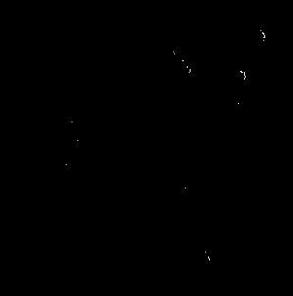
    for m in range(x):
        for n in range(y):
            if pixel_citra[m,n] < nilai_ambang:
                pixel_citra[m,n] = 0
            else:
                pixel_citra[m,n] = 255

    citra_disimpan = 'D:/IMAGE PROCESSING/MATERIAL/Biner/citra_biner_' + str(nilai_ambang)+'.jpg'
    citra.save(citra_disimpan)

# Press the green button in the gutter to run the script.
if __name__ == '__main__':
    citra_biner(50)
```

Hasil Output:

Mengubah citra grayscale menjadi citra biner

Threshold = 50	Threshold = 128
	
Threadhold = 200	Threshold = 230
	

Silakan Ubah nilai Threshold sesuai dengan nilai yang telah ditentukan.

Pertanyaan :

1. Amati perbedaan dari nilai threshold gambar diatas pada citra biner. Jelaskan perbedaannya dan alasannya mengapa setiap output yang dihasilkan dengan nilai threshold yang berbeda menghasilkan gambar yang berbeda !

Jawaban :

-Perbedaan

dalam setiap gambar berbeda dalam pencerahan nya setiap ditambah nya threshold dari 50 ke 250 maka gambar akan menjadi gelap

-alasan nya

Ada nya Nilai intensitas citra yang lebih dari atau sama dengan nilai threshold akan diubah menjadi putih (1) sedangkan nilai intensitas citra yang kurang dari nilai threshold akan diubah menjadi hitam (0) Sehingga keluaran dari hasil thresholding adalah berupa citra biner.

2. Buatlah program mengubah citra grayscale ke binary dengan menggunakan fungsi **cv2.threshold()** yang ada di python !

Matlab

```
CitraBiner.m +  
1 - function fig = CitraBiner(T)  
2 -     citra = imread('D:/IMAGE PROCESSING/MATERIAL/lena_grayscale.jpg');  
3 -     fig = figure;  
4 -     [x,y]=size(citra);  
5 -     citra_biner=zeros(x,y);  
6 -     for m=1:x  
7 -         for n=1:y  
8 -             if(citra(m,n) > T)  
9 -                 citra_biner(m,n) = 1;  
10 -            else  
11 -                citra_biner(m,n) = 0;  
12 -            end  
13 -        end  
14 -    end  
15 -    imshow(citra_biner);  
16 - end
```

Eksekusi fungsi CitraBiner pada Matlab :

```
>> fig = CitraBiner(50);
```

Hasil Output

Threshold = 50	Threshold = 128
	
Threshold = 200	Threshold = 230

Silakan Ubah nilai Threshold sesuai dengan nilai yang telah ditentukan.

Pertanyaan :

1. Amati perbedaan dari nilai threshold gambar diatas pada citra biner. Jelaskan perbedaannya dan alasannya mengapa setiap output yang dihasilkan dengan nilai threshold yang berbeda menghasilkan gambar yang berbeda !
2. Buatlah program mengubah citra grayscale ke binary dengan menggunakan fungsi **im2bw()** yang ada di matlab !

2. CITRA NEGATIVE

Dengan mengubah citra grayscale menjadi citra negative dengan mengurangi nilai intensitas pixel dari nilai keabuan maksimum.

$$f(x, y)' = 255 - f(x, y)$$

Python

```
import cv2

citra = cv2.imread('D:/IMAGE PROCESSING/MATERIAL/lena_grayscale.jpg')
citra_negative = 255 - citra
cv2.imshow('Image Asli', citra)
cv2.imshow('Image Negative', citra_negative)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Hasil eksekusi program :



Pertanyaan

Silakan Jelaskan setiap baris program serta tampilkan ScreenShot hasil eksekusi dari program diatas.

`import cv2`

berfungsi untuk memanggil package atau modul cv2

`citra = cv2.imread('D:/Rasheed/image processing/lena_grayscale.jpg')`

berfungsi untuk memanggil gambar dari dalam folder kita ke variable

`citra_negative = 255 - citra`

berfungsi untuk mengatur keabuan grayscale yang terdapat pada gambar

`cv2.imshow('image asli',citra)`

`cv2.imshow ('image negative',citra_negative)`

berfungsi untuk menampilkan gambar

`cv2.waitKey(0)`

untuk mempertahankan window agar tetap menampilkan gambar.

`cv2.destroyAllWindows()`

berfungsi untuk menutup dari windows

File Edit Selection View Go Run Terminal Help

Get Started uts1(3).py uts1(2).py

image proccesing > uts1(2).py > ...

```
1 import cv2
2 citra = cv2.imread('D:/Rasheed/image proccesing/lena_grayscale.jpg')
3 citra_negative = 255 - citra
4 cv2.imshow('image asli',citra)
5 cv2.imshow('image negative',citra_negative)
6 cv2.waitKey()
7 cv2.destroyAllWindows()
```

image asli

image negative

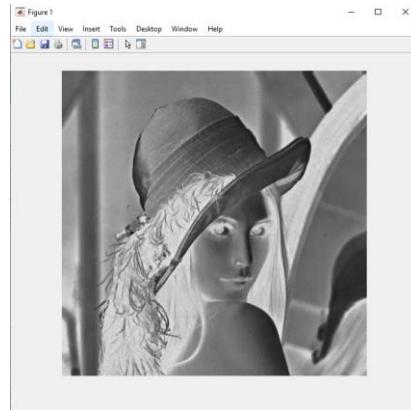
uts1(2).py - Rasheed - Visual Studio Code

iMatlab

```
1 function fig = CitraNegative(A)
2 -     fig = figure;
3 -     [x,y]=size(A);
4 -     for m=1:x
5 -         for n=1:y
6 -             image_negative(m,n)= 255-A(m,n);
7 -         end
8 -     end
9 -     imshow(image_negative);
10 -    end
```

Hasil Output program :

```
>> B = imread('D:/IMAGE PROCESSING/MATERIAL/lena_grayscale.jpg');
>> fig = CitraNegative(B)
```



Pertanyaan

Silakan Jelaskan setiap baris program serta tampilkan ScreenShot hasil eksekusi dari program diatas.

3. PENCERAHAN CITRA

Melakukan dengan cara menambahkan/mengurangkan konstanta nilai dari suatu pixel dengan persamaan sebagai berikut :

$$f'(x, y) = f(x, y) + b$$

Dengan melakukan operasi clipping pada nilai pixel dicitra :

$$f'(x, y) = \begin{cases} 255, & f(x, y) > 255 \\ f(x, y), & 0 \leq f(x, y) \leq 255 \\ 0, & f(x, y) < 0 \end{cases}$$

Python

```
#Reference : https://github.com/imambungo/operasi-citra/blob/master/citra_pencerahan.py
from PIL import Image

def clipping(intensitas):
    if intensitas < 0:
        return 0
    if intensitas > 255:
        return 255
    return intensitas

def pencerahaan_image(nilai_intensitas):
    # gunakan JPG image
    citra = Image.open('D:/IMAGE PROCESSING/MATERIAL/flowers.jpg')
    citra_cerah = citra.load()

    x = citra.size[0]
    y = citra.size[1]

    for m in range(x):
        for n in range(y):
            R = clipping(citra_cerah[m, n][0] + nilai_intensitas)
            G = clipping(citra_cerah[m, n][1] + nilai_intensitas)
            B = clipping(citra_cerah[m, n][2] + nilai_intensitas)
            citra_cerah[m, n] = (R, G, B)

    citra.save('D:/IMAGE PROCESSING/MATERIAL/citra_cerah_2.png')
```

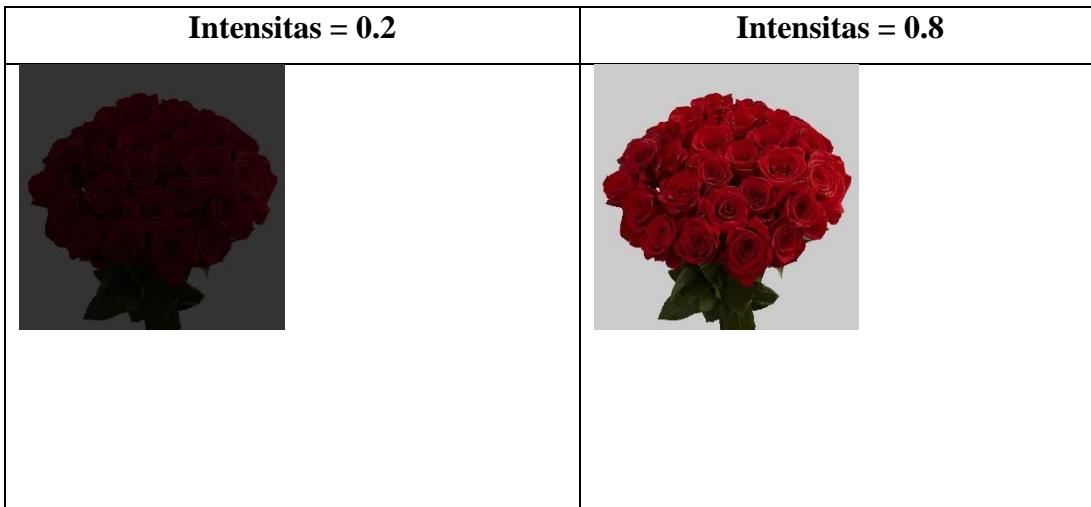
Pertanyaan

1. Buat baris program untuk dapat menampilkan hasil pencerahan dengan nilai intensitas 80 dan -120. Screenshot baris program dan hasil output program yang dijalankan.

Intensitas = 80	Intensitas = -120
	

Amati dan jelaskan perbedaan antara kedua nilai intentitas serta alasan mengapa hasil output yang dikeluarkan berbeda.

2. Buatlah program dan jelaskan setiap baris program yang dibuat untuk pencerahan citra dengan menggunakan **imadjust()** yang terdapat pada matlab dengan nilai intentitas 0.2 dan 0.8



Matlab

```
|function B=Clipping(A,C)
|for m=1:size(A,1)
|    for n=1:size(A,2)
|        B(m,n) = A(m,n) + C;
|        if A(m,n) < 0
|            B(m,n) = 0;
|        elseif A(m,n) > 255
|            B(m,n) = 255;
|        else
|            B(m,n)= A(m,n);
|        end
|    end
|end
```

Pertanyaan :

1. Buat baris program untuk dapat menampilkan hasil pencerahan dengan nilai intentitas 80 dan -120. Screenshot baris program dan hasil output program yang dijalankan.

Intensitas = 80	Intensitas = -120

Amati dan jelaskan perbedaan antara kedua nilai intentitas serta alasan mengapa hasil output yang dikeluarkan berbeda.

2. Buatlah program dan jelaskan setiap baris program yang dibuat untuk pencerahan citra dengan menggunakan **ImageEnhance.Brightness()** yang terdapat pada library Pillow(PIL) dengan nilai intentitas 80 dan -120

Intensitas = 80	Intensitas = -120

4. KONVERSI CITRA WARNA KE GRayscale

Mengkonversi citra dengan kanal RGB menjadi citra grayscale dengan persamaan :

$$Y = 0.299R + 0.587G + 0.144B$$

Python

```
import matplotlib.pyplot as plt
import matplotlib.image as image

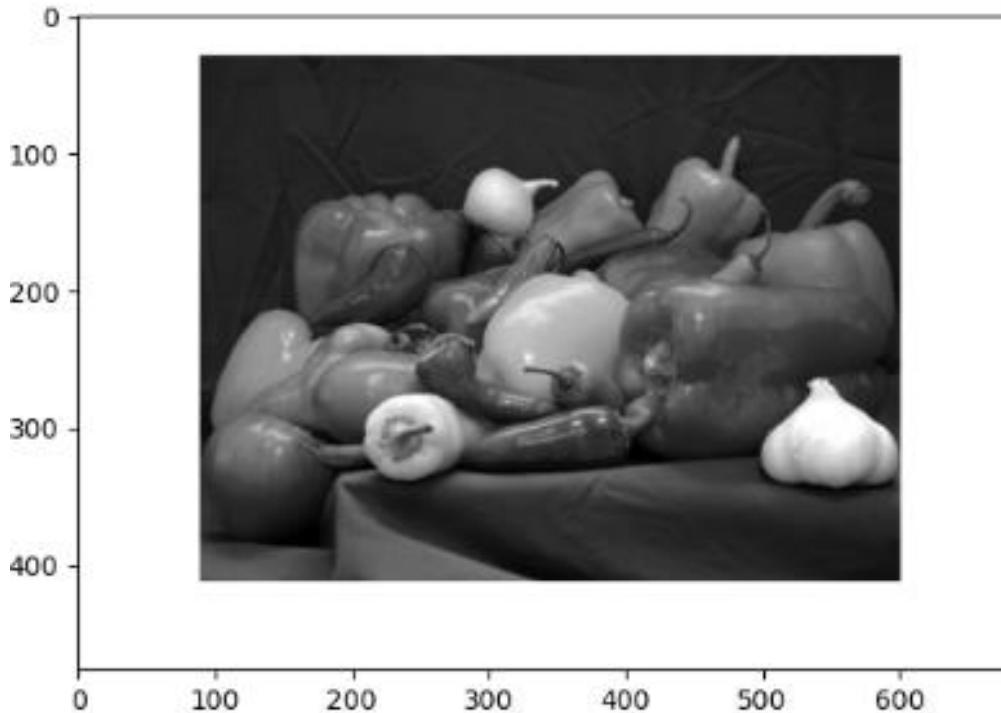
citra = image.imread('D:/IMAGE PROCESSING/MATERIAL/RGB_Citra.png')

def rgb2gray(citra):
    R = citra[:, :, 0]
    G = citra[:, :, 1]
    B = citra[:, :, 2]
    citra_gray = R * 0.2989 + G * 0.5870 + B * 0.1141

    return citra_gray

gray = rgb2gray(citra)
plt.imshow(gray, cmap='gray')
plt.show()
```

Hasil Output:



Pertanyaan

1. Jelaskan setiap baris program diatas !

```
import matplotlib.pyplot as plt  
import matplotlib.image as image
```

Dalam Python kita bisa dengan mudah melakukan visualisasi data dengan library **Matplotlib**. **matplotlib.pyplot** adalah kumpulan fungsi yang membuat beberapa perubahan pada gambar: misalnya, membuat gambar, membuat area plot dalam gambar, menambah label di plot dan lainnya. Biasanya untuk mempermudah secara umum **matplotlib.pyplot** disingkat menjadi **plt**.

```
def rgb2gray(citra):  
    R = citra[:, :, 0]  
    G = citra[:, :, 1]  
    B = citra[:, :, 2]  
    citra_gray = R * 0.2989 + G * 0.5870 + B * 0.1141
```

menghitung nilai grayscale untuk lebih di mudah menghitung manual dan mudah di ingat digunakan persamaan yang kedua

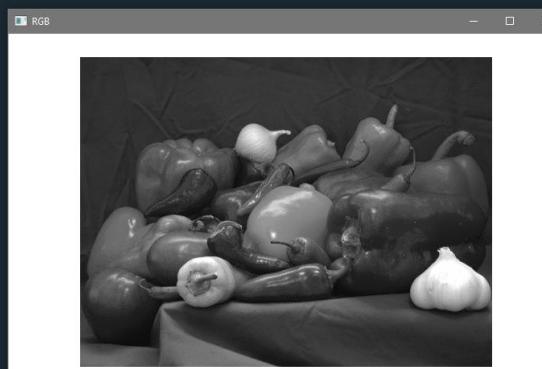
```
citra = image.imread('D:/Rasheed/image proccesing/RGB_Citra.png')
```

Imread digunakan untuk membaca image dari file grafis yang hasilnya berupa matrik dan disimpan dalam sebuah variabel citra

2. Buatlah program untuk mengkonvert citra RGB ke gray dengan menggunakan fungsi **cv2.COLOR_RGB2GRAY** di Python. (Screenshot kode program serta jelaskan setiap kode program yang anda buat dan tampilkan hasil outputnya)

```
image proccesing > cv2rgb.py > ...
```

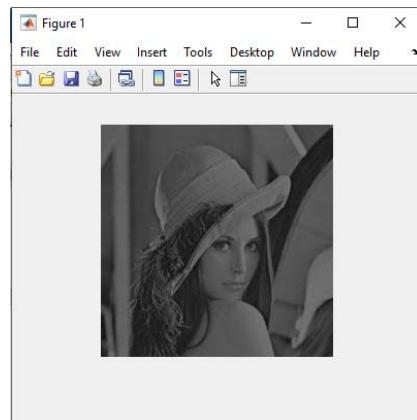
```
1 import cv2  
2  
3 image = cv2.imread('D:/Rasheed/image proccesing/RGB_Citra.png')  
4  
5 gambar = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)  
6  
7 cv2.imshow('RGB',gambar)  
8 cv2.waitKey(0)
```



Matlab

```
>> B = imread('D:/IMAGE PROCESSING/MATERIAL/lena_warna.jpeg');
>> gray = uint8(0.2989 * double(B(:,:,1)) + 0.5870 * double(B(:,:,2)) + 0.1141 * double(B(:,:,3)));
>> imshow(gray)
```

Hasil Ouput :



Pertanyaan

1. Jelaskan setiap baris program diatas
2. Buatlah program untuk mengkonvert citra RGB ke gray dengan menggunakan fungsi **rgb2gray** di matlab. (Screenshot kode program serta jelaskan setiap kode program yang anda buat dan tampilkan hasil outputnya)

5. OPERASI ARITMETIK

5.1 PENJUMLAHAN CITRA

Operasi aritmatika penjumlahan pada citra :

$$C(x, y) = A(x, y) + B(x, y)$$

Python

```
from PIL import Image

def penambahan_citra(citra_A, citra_B, citra_hasil):
    CITRA_A = Image.open(citra_A)
    PIXEL_A = CITRA_A.load()

    CITRA_B = Image.open(citra_B)
    PIXEL_B = CITRA_B.load()

    ukuran_horizontal = CITRA_A.size[0]
    ukuran_vertikal = CITRA_A.size[1]

    CITRA_HASIL = Image.new("RGB", (ukuran_horizontal, ukuran_vertikal))
    PIXEL_HASIL = CITRA_HASIL.load()

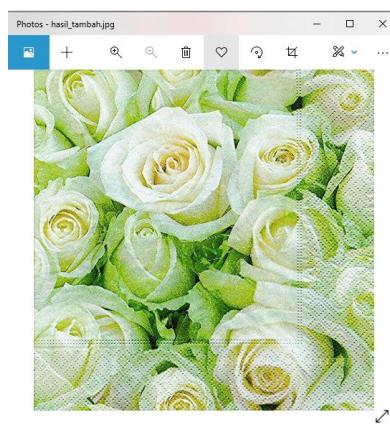
    for x in range(ukuran_horizontal):
        for y in range(ukuran_vertikal):
            R = PIXEL_A[x, y][0] + PIXEL_B[x, y][0]
            G = PIXEL_A[x, y][1] + PIXEL_B[x, y][1]
            B = PIXEL_A[x, y][2] + PIXEL_B[x, y][2]
            PIXEL_HASIL[x, y] = (R, G, B)

            # jika terdapat perbedaan antara pixel A dan pixel B,
            # gunakan pixel B
            if R > 255 or G > 255 or B > 255:
                PIXEL_HASIL[x, y] = PIXEL_B[x, y]

    CITRA_HASIL.save(citra_hasil)

if __name__ == '__main__':
    penambahan_citra('D:/IMAGE PROCESSING/MATERIAL/mawar1.jpg',
                      'D:/IMAGE PROCESSING/MATERIAL/rose2.jpg',
                      'D:/IMAGE PROCESSING/MATERIAL/hasil_tambah.jpg')
```

Hasil output program :



Pertanyaan :

1. Buatlah program penambahan image seperti dibawah ini !

```
import cv2
import numpy as np
from PIL import Image

def penambahan_citra(citra_A, citra_B):
    A = cv2.imread(citra_A)
    B = cv2.imread(citra_B)

    height_A, width_A = A.shape[:2]
    citra_hasil = np.ones([height_A, width_A, 3], np.uint8)
    for x in range(height_A):
        for y in range(width_A):
            citra_hasil[x, y] = A[x, y] + B[x, y]
            if (citra_hasil[x, y] > 255).all():
                citra_hasil[x, y] = 255

    cv2.imshow('Hasil Output Tambah Citra', citra_hasil)

if __name__ == '__main__':
    citra_A = 'D:/IMAGE PROCESSING/MATERIAL/mawar1.jpg'
    citra_B = 'D:/IMAGE PROCESSING/MATERIAL/rose2.jpg'

    penambahan_citra(citra_A, citra_B)

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

Screenshot hasil eksekusi untuk program diatas !



```
image processing > nomer5.py > ...
  9     height_A, width_A = A.shape[:2]
10     citra_hasil = np.ones([height_A, width_A, 3], np.uint8)
11     for x in range(height_A):
12         for y in range(width_A):
13             citra_hasil[x, y] = A[x, y] + B[x, y]
14             if (citra_hasil[x, y] > 255).all():
15                 citra_hasil[x, y] = 255
16
17     cv2.imshow('hasil Output Tambah Citra', citra_hasil)
18
19 if __name__ == '__main__':
20     citra_A = 'D:/Rasheed/image procesing/mawar1.jpg'
21     citra_B = 'D:/Rasheed/image procesing/rose2.jpg'
22
23     penambahan_citra(citra_A, citra_B)
24
25 cv2.waitKey(0)
26 cv2.destroyAllWindows()
```

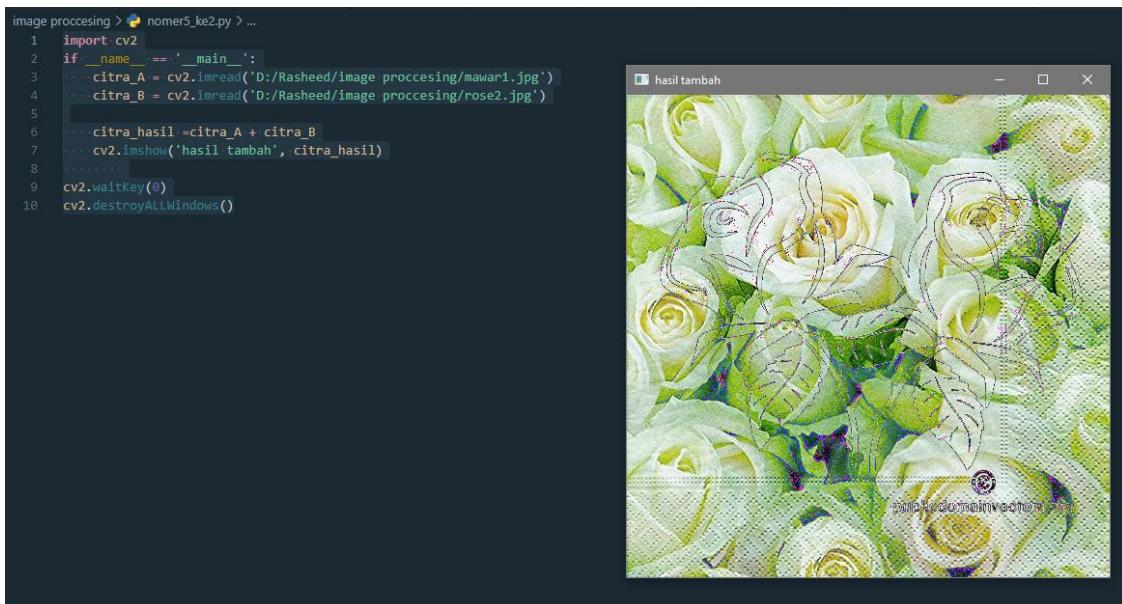
2. Buatlah program seperti dibawah ini

```
import cv2
if __name__ == '__main__':
    citra_A = cv2.imread('D:/IMAGE PROCESSING/MATERIAL/mawar1.jpg')
    citra_B = cv2.imread('D:/IMAGE PROCESSING/MATERIAL/rose2.jpg')

    citra_hasil = citra_A + citra_B
    cv2.imshow('Hasil Tambah', citra_hasil)

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

Screenshot hasil eksekusi untuk program diatas !

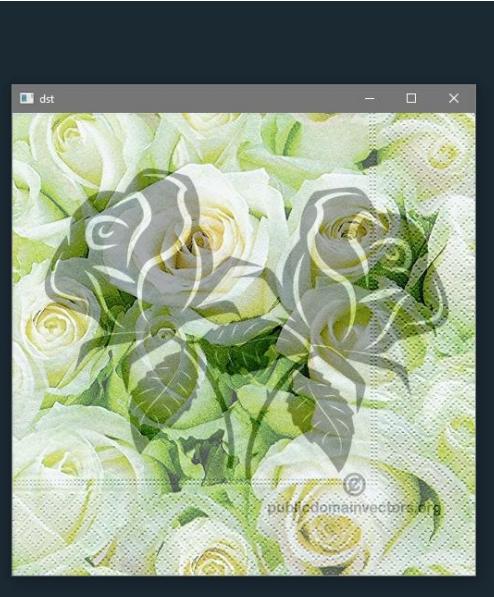


Jelaskan perbedaan hasil eksekusi program pada no.1, no.2 dengan program diatasnya yang masih dalam sub-bab penjumlahan citra serta alasan perbedaan tersebut !

3. Buatlah program penjumlahan citra dengan menggunakan fungsi **cv2.addWeighted()** dengan menggunakan python ! (Screenshot program dan hasil eksekusinya)

```
image proccesing > penjumlahcitra_bag3.py > ...
```

```
1 import sys
2 import cv2
3 from PIL import Image, ImageOps
4 img1 = cv2.imread('D:/Rasheed/image proccesing/mawar1.jpg', 1)
5 img2 = cv2.imread('D:/Rasheed/image proccesing/rose2.jpg', 1)
6 dst = cv2.addWeighted(img1,0.3,img2,0.7,0)
7 cv2.imshow('dst',dst)
8 cv2.waitKey(0)
9 cv2.destroyAllWindows()
```



Matlab

Buat fungsi JumlahCitra seperti berikut ini di matlab

```
function fig=JumlahCitra(A,B)
    fig=figure;
    [h w]=size(A);
    [k l]=size(B);
    if(h==k) && (w==l)
        for baris=1:h
            for kolom=1:w
                C(baris,kolom)= A(baris,kolom)+B(baris,kolom);
            end
        end
        imshow(C);
    end
```

Pertanyaan :

1. Buatlah program untuk melakukan eksekusi fungsi diatas dengan menggunakan file **mawar1.jpg** dan **rose2.jpg** !
2. Buatlah program dan jelaskan setiap baris program untuk melakukan penjumlahan citra dengan menggunakan **fungsi imadd()** di matlab. (Screenshot program dan hasil eksekusinya)

5.2 PENGURANGAN CITRA

Operasi aritmatika pengurangan pada citra :

$$C(x, y) = A(x, y) - B(x, y)$$

Python

```
# Referensi : https://github.com/imambungo/operasi-citra

from PIL import Image

def pengurangan_citra(citra_A, citra_B, citra_hasil):
    CITRA_A = Image.open(citra_A)
    PIXEL_A = CITRA_A.load()

    CITRA_B = Image.open(citra_B)
    PIXEL_B = CITRA_B.load()

    ukuran_horizontal = CITRA_A.size[0]
    ukuran_vertikal = CITRA_A.size[1]

    CITRA_HASIL = Image.new("RGB", (ukuran_horizontal, ukuran_vertikal))
    PIXEL_HASIL = CITRA_HASIL.load()

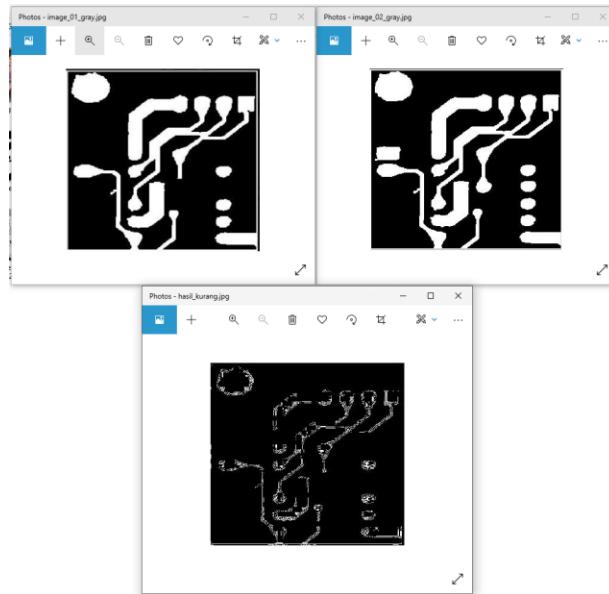
    for x in range(ukuran_horizontal):
        for y in range(ukuran_vertikal):
            R = PIXEL_A[x, y][0] - PIXEL_B[x, y][0]
            G = PIXEL_A[x, y][1] - PIXEL_B[x, y][1]
            B = PIXEL_A[x, y][2] - PIXEL_B[x, y][2]
            PIXEL_HASIL[x, y] = (R, G, B)

            # jika terdapat perbedaan antara pixel A dan pixel B,
            # gunakan pixel B
            if R > 0 or G > 0 or B > 0:
                PIXEL_HASIL[x, y] = PIXEL_B[x, y]

    CITRA_HASIL.save(citra_hasil)

if __name__ == '__main__':
    pengurangan_citra('D:/IMAGE PROCESSING/MATERIAL/image_01_gray.jpg',
                      'D:/IMAGE PROCESSING/MATERIAL/image_02_gray.jpg',
                      'D:/IMAGE PROCESSING/MATERIAL/hasil_kurang.jpg')
```

Hasil Output program :



Pertanyaan :

1. Silakan buat program seperti di bawah ini

```
import cv2

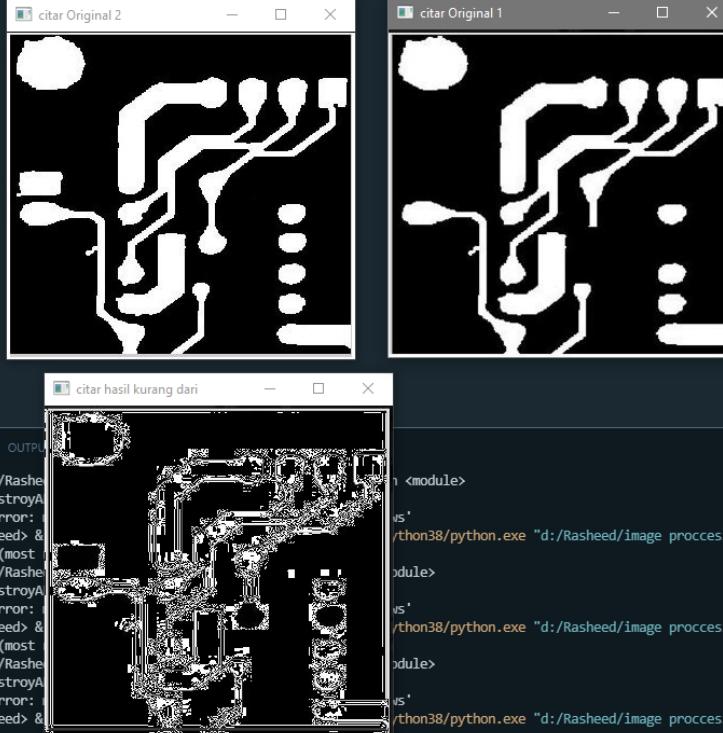
if __name__ == '__main__':
    citra_A = cv2.imread('D:/IMAGE PROCESSING/MATERIAL/image_01_gray.jpg')
    citra_B = cv2.imread('D:/IMAGE PROCESSING/MATERIAL/image_02_gray.jpg')

    citra_kurang = citra_A - citra_B
    cv2.imshow('Citra Original 1', citra_A)
    cv2.imshow('Citra Original 2', citra_B)
    cv2.imshow('Citra Hasil Kurang', citra_kurang)

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

Apa yang didapatkan dari program diatas (Screenshot hasil output program)!

```
image procesing > nomer5_pengurangan_citra.py > ...
1  import cv2
2
3  if __name__ == '__main__':
4      citra_A = cv2.imread('D:/Rasheed/image procesing/image_01_gray.jpg')
5      citra_B = cv2.imread('D:/Rasheed/image procesing/image_02_gray.jpg',)
6
7      citra_kurang = citra_A - citra_B
8      cv2.imshow('citar Original 1', citra_A)
9      cv2.imshow('citar Original 2', citra_B)
10     cv2.imshow('citar hasil kurang dari', citra_kurang)
11
12    cv2.waitKey(0)
13    cv2.destroyAllWindows()
14
```



```
File "d:/Rasheed/image procesing/nomer5.py"
cv2.destroyAllWindows()
AttributeError:
PS D:\Rasheed> &
Traceback (most recent call last):
File "d:/Rasheed/image procesing/nomer5.py", line 14, in <module>
cv2.destroyAllWindows()
AttributeError:
PS D:\Rasheed> &
Traceback (most recent call last):
File "d:/Rasheed/image procesing/nomer5.py", line 14, in <module>
cv2.destroyAllWindows()
AttributeError:
PS D:\Rasheed> &
Traceback (most recent call last):
File "d:/Rasheed/image procesing/nomer5.py", line 14, in <module>
cv2.destroyAllWindows()
```

Amati dan jelaskan perbedaan antara hasil program pada soal no.1 dengan program diatasnya. Serta jelaskan alasannya mengapa berbeda.:

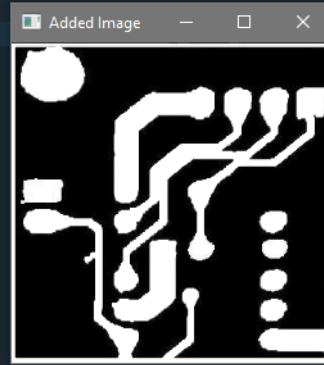
-perbedaan nya

Nomer 1 adanya fungsi yang di mana terelebih dahulu RGB dari pada gambar yang sudah di variable oleh pixel_A dan PIXEL_B yang di mana hasil RGB dari gambar yang sudah di variable kan itu di tampilkan

Nomer 2 Dengan piksel citra C adalah hasil pengurangan intensitas piksel citra A dengan citra B dan di tampilkan nya hasil dari pengurangan intensitas pixel tersebut

2. Buatlah program aritmatika pengurangan citra dengan menggunakan fungsi **cv2.add()** pada Python ! (Screenshot program dan hasilnya)

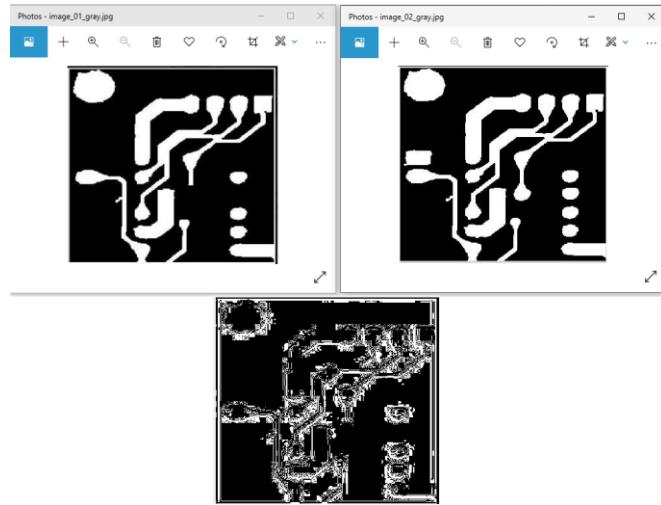
```
image proccesing > pengurangan_citra2.py > ...
1  import cv2
2
3  imagePortrait = cv2.imread('D:/Rasheed/image proccesing/image_01_gray.jpg', cv2.IMREAD_COLOR)
4  imageLogo = cv2.imread('D:/Rasheed/image proccesing/image_02_gray.jpg', cv2.IMREAD_COLOR)
5
6  #Need to set the same size for both Loaded images.
7  #Both images size should be the same before using the add method
8  imagePortrait = cv2.resize(imagePortrait, (250,250))
9  imageLogo = cv2.resize(imageLogo, (250,250))
10
11 #add method
12 addedImage = cv2.add(imagePortrait, imageLogo)
13 cv2.imshow("Added Image", addedImage)
14 cv2.waitKey()
15 cv2.destroyAllWindows()
```



Matlab

```
function fig=KurangCitra(A,B)
    fig=figure;
    [h w]=size(A);
    [k l]=size(B);
    if(h==k)&&(w==l)
        for baris=1:h
            for kolom=1:w
                C(baris,kolom)= max(A(baris,kolom)- B(baris,kolom),0);
                if (C(baris,kolom) ~= 0)
                    C(baris,kolom)= 255;
                end
            end
        end
        imshow(C);
    end
```

Hasil Output program :



Pertanyaan :

1. Silakan buat program seperti di bawah ini

```
>> A = imread('D:/IMAGE PROCESSING/MATERIAL/image_01_gray.jpg');
>> B = imread('D:/IMAGE PROCESSING/MATERIAL/image_02_gray.jpg');
>> C = A - B;
>> imshow(A);
>> imshow(B);
>> imshow(C);
```

Apa yang didapatkan dari program diatas (Screenshot hasil output program)!

Amati dan jelaskan perbedaan antara hasil program pada soal no.1 dengan program diatasnya. Serta jelaskan alasannya mengapa berbeda.

2. Buatlah program aritmatika pengurangan citra dengan menggunakan fungsi **imsubtract()** pada matlab ! (Screenshot program dan hasilnya)

5.3 PERKALIAN CITRA

Operasi aritmatika untuk perkalian citra :

$$C(x, y) = A(x, y) * B(x, y)$$

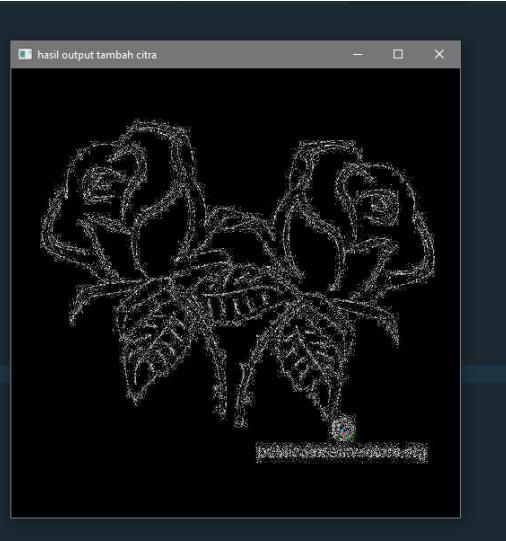
Python

```
1 import cv2
2 import numpy as np
3
4 def perkalian_citra(citra_A, citra_B):
5     A = cv2.imread(citra_A)
6     B = cv2.imread(citra_B)
7
8     height_A, width_A = A.shape[:2]
9     citra_hasil = np.ones([height_A, width_A, 3], np.uint8)
10    for x in range(height_A):
11        for y in range(width_A):
12            citra_hasil[x, y] = A[x, y] * B[x, y]
13
14    cv2.imshow('Hasil Output Tambah Citra', citra_hasil)
15
16    if __name__ == '__main__':
17        citra_A = 'D:/IMAGE PROCESSING/MATERIAL/mawar1.jpg'
18        citra_B = 'D:/IMAGE PROCESSING/MATERIAL/mawar1.jpg'
19
20        perkalian_citra(citra_A, citra_B)
21
22    cv2.waitKey(0)
```

Pertanyaan :

1. Screenshot hasil eksekusi dari program diatas! dan jelaskan apa yang anda dapatkan !

```
image procesing > ⚡ perkalian.py > ...
1  import cv2
2  import numpy as np
3
4  def perkalian_citra(citra_A, citra_B):
5      A = cv2.imread(citra_A)
6      B = cv2.imread(citra_B)
7
8      height_A, width_A = A.shape[:2]
9      citra_hasil = np.ones([height_A, width_A, 3], np.uint8)
10     for x in range (height_A):
11         for y in range (width_A):
12             citra_hasil[x, y] = A[x, y] * B[x, y]
13
14     cv2.imshow('hasil output tambah citra', citra_hasil)
15
16    if __name__ == '__main__':
17        citra_A = 'D:/Rasheed/image procesing/mawar1.jpg'
18        citra_B = 'D:/Rasheed/image procesing/mawar1.jpg'
19
20        perkalian_citra(citra_A, citra_B)
21
22    cv2.waitKey(0)
```



Yang saya dapatkan adalah hasil perkalian antar gambar CITRA_A dan CITRA_B

2. Buatlah program seperti dibawah ini !

```
import cv2
if __name__ == '__main__':
    citra_A = cv2.imread('D:/IMAGE PROCESSING/MATERIAL/mawar1.jpg')
    citra_B = cv2.imread('D:/IMAGE PROCESSING/MATERIAL/mawar1.jpg')

    citra_hasil = citra_A * citra_B
    cv2.imshow('Hasil Tambah', citra_hasil)

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

Apa yang didapatkan dari program diatas (Screenshot hasil output program)!

The screenshot shows a Visual Studio Code interface with a dark theme. On the left is the Explorer sidebar listing various Python files and image files. The main editor area contains a Python script named 'perkalian2.py' which performs element-wise multiplication of two grayscale images. To the right of the editor is a preview window titled 'citar hasil perkalian dari' displaying the resulting grayscale image, which is a composite of two flower images from the input files.

```
import cv2
if __name__ == '__main__':
    citra_A = cv2.imread('D:/Rasheed/image procesing/image_01_gray.jpg')
    citra_B = cv2.imread('D:/Rasheed/image procesing/image_02_gray.jpg')

    citra_kurang = citra_A * citra_B
    cv2.imshow('citar hasil perkalian dari', citra_kurang)

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

3. Buatlah program dan jelaskan setiap baris program ddengan melakukan perkalian citra dengan menggunakan **cv2.multiply()** pada matlab. Screenshot program dan hasil output program yang didapatkan

The screenshot shows the Visual Studio Code interface. The left sidebar displays a file tree under 'RASHEED' containing various Python files and image files. The main editor window shows the code for 'multiply.py'. The code imports cv2, defines a function 'multiply_demo', and reads two images ('mawar1.jpg' and 'mawar1.jpg') to perform a multiplication operation. The output window shows a black and white image of two roses.

```

File Edit Selection View Go Run Terminal Help
EXPLORER RASHEED
divide.py
flipping.py
flowers.jpg
from_PIL_import_Image.py
fruits.jpeg
image_01_gray.jpg
image_02_gray.jpg
lenna_grayscale.jpg
lenna_wmns.jpeg
mawar1.jpg
multiply.py
nomer1_1.py
nomer1_2.py
nomer5_pengurangan_citra.py
nomers.py
nomero_1.py
operasi_and.py
operasi_bitwiseand.py
operasi_not.py
operasi_or.py
pembagian_citra.py
pembagian.py
pengkalian_image.py
pengurangan_citra2.py
pengurangan_skalar.py
pengurangan_skalar2.py
penjumlahan_skalar.py
penjumlahan_skalar_bagi3.py
penjumlahan_citra_bagi3.py
perkalian_skalar.py
ut1(2).py
ut1(3).py
perkalian2.py
multiply.py
cv2rgb.py
nomer5.py
nomer5_pengurangan_citra.py
pengurangan_citra2.py
image processing > multiply.py ...
1 import cv2 as cv
2
3 def multiply_demo(src1, src2):
4     ret = cv2.multiply(src1, src2)
5     cv2.imshow("multiply_demo", ret)
6     cv2.waitKey(0)
7     cv2.destroyAllWindows()
8
9 if __name__ == "__main__":
10    src1 = cv2.imread("D:/Rasheed/image proccesing/mawar1.jpg")
11    src2 = cv2.imread("D:/Rasheed/image proccesing/mawar1.jpg")
12    multiply_demo(src1, src2)

```

Amati dan jelaskan perbedaan antara hasil program sebelumnya pada sub-bab perkalian citra dengan program diatasnya. Serta jelaskan alasannya mengapa berbeda.

No. 1 adalah perkalian menggunakan looping for

No. 2 adalah langsung dengan mengalikan citra A dan B

Sedangkan No. 3 menggunakan cv2.multiply sehingga program lebih sederhanan Maka harus diketahui yang mana yang Anda butuhkan

Matlab

```

function fig=PerkalianCitra(A,B)
    fig=figure;
    [h w]=size(A);
    [k l]=size(B);
    if(h==k) && (w==l)
        for baris=1:h
            for kolom=1:w
                C(baris,kolom)= A(baris,kolom)* B(baris,kolom);
            end
        end
        imshow(C);
    end
end

```

Pertanyaan :

1. Screenshot hasil eksekusi dari proram diatas ! jelaskan apa yang ada dapatkan !

2. Buatlah program untuk melakukan perkalian citra dengan menggunakan fungsi **immmultiply()** yang ada pada matlab. Screenshot program dan hasil eksekusi.

Amati dan jelaskan perbedaan antara hasil program sebelumnya pada sub-bab perkalian citra dengan program diatasnya. Serta jelaskan alasannya mengapa berbeda

5.4 PEMBAGIAN CITRA

Operasi aritmatika pembagian citra : $C(x,y) = A(x,y) / B(x,y)$

Python

Pertanyaan :

1. Buatlah program dan jelaskan setiap baris program dengan melakukan operasi citra dengan menggunakan operasi aritmatik pembagian ! Screenshot program dan hasil output program yang didapatkan.

```
import cv2
```

berfungsi untuk memanggil package atau modul cv2

```
citraA = cv2.imread('D:/Rasheed/image proccesing/mawar1.jpg')  
citraB = cv2.imread('D:/Rasheed/image proccesing/rose2.jpg')
```

berfungsi untuk memanggil gambar dari dalam folder kita ke variable

```
citra_hasil = citraA / citraB
```

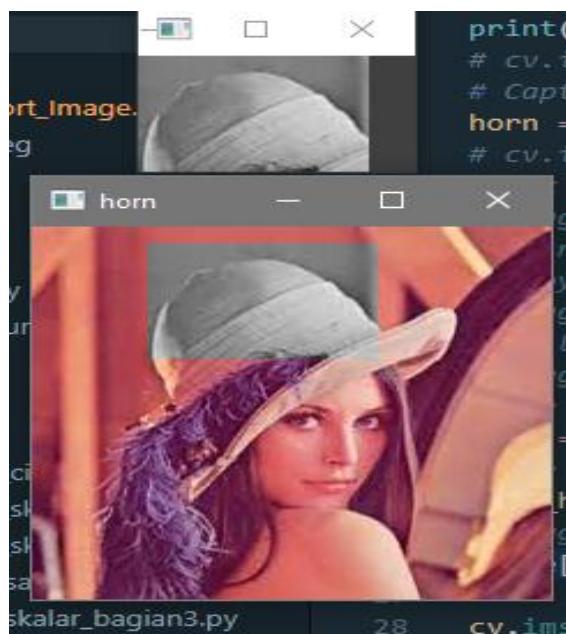
pembagian antara citra

```
cv2.imshow('Hasil Pembagian', citra_hasil)
```

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

menampilkan hasil citra dari peda pembagian citra yang di lakukan di atas

2. Buatlah program dan jelaskan setiap baris program ddengan melakukan pembagian citra dengan menggunakan **cv2.divide()** pada matlab. Screenshot program dan hasil output program yang didapatkan



Matlab

Pertanyaan :

1. Buatlah program dan jelaskan setiap baris program dengan melakukan operasi citra dengan menggunakan operasi aritmatik pembagian ! Screenshot program dan hasil output program yang didapatkan.
2. Buatlah program dan jelaskan setiap baris program dengan melakukan pembagian citra dengan menggunakan **imdivide()** pada matlab. Screenshot program dan hasil output program yang didapatkan

5.5 PENJUMLAHAN CITRA DENGAN SKALAR

Operasi aritmatika penjumlahan citra dengan scalar :

$$B(x,y) = A(x,y) + C$$

Python

```
import cv2
import numpy as np

def penambahan_citra_skalar(citra_A, nilai_skalar):
    A = cv2.imread(citra_A)

    height_A, width_A = A.shape[:2]
    citra_hasil = np.ones([height_A, width_A, 3], np.uint8)
    for x in range(height_A):
        for y in range(width_A):
            citra_hasil[x, y] = A[x, y] + nilai_skalar

    cv2.imshow('Hasil Output Tambah Citra Skalar', citra_hasil)

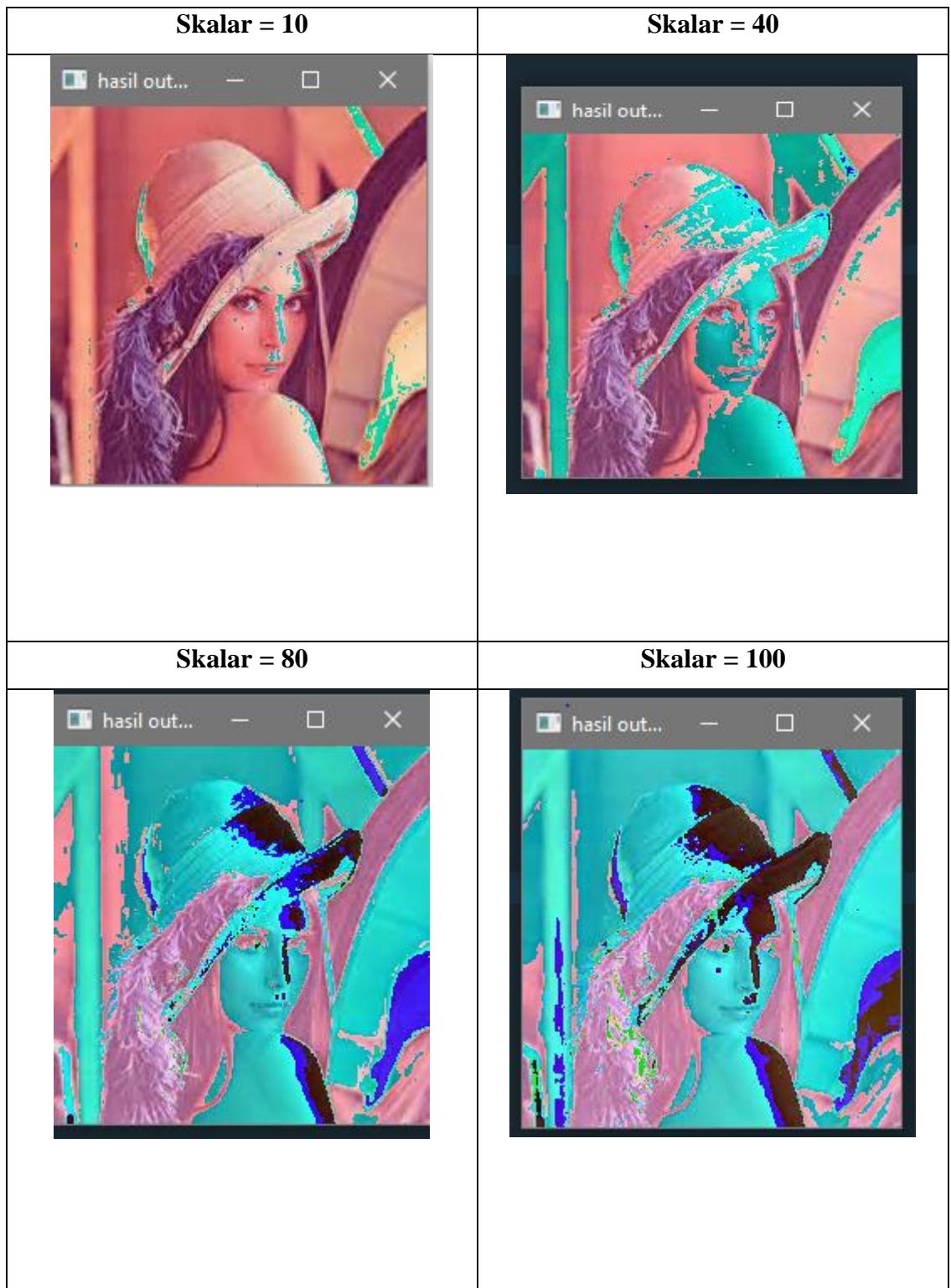
if __name__ == '__main__':
    citra_A = 'D:/IMAGE PROCESSING/MATERIAL/lena_warna.jpeg'

    penambahan_citra_skalar(citra_A, 10)

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

Pertanyaan :

1. Screenshot hasil output pada program diatas dengan nilai scalar 10, 40, 80, 100



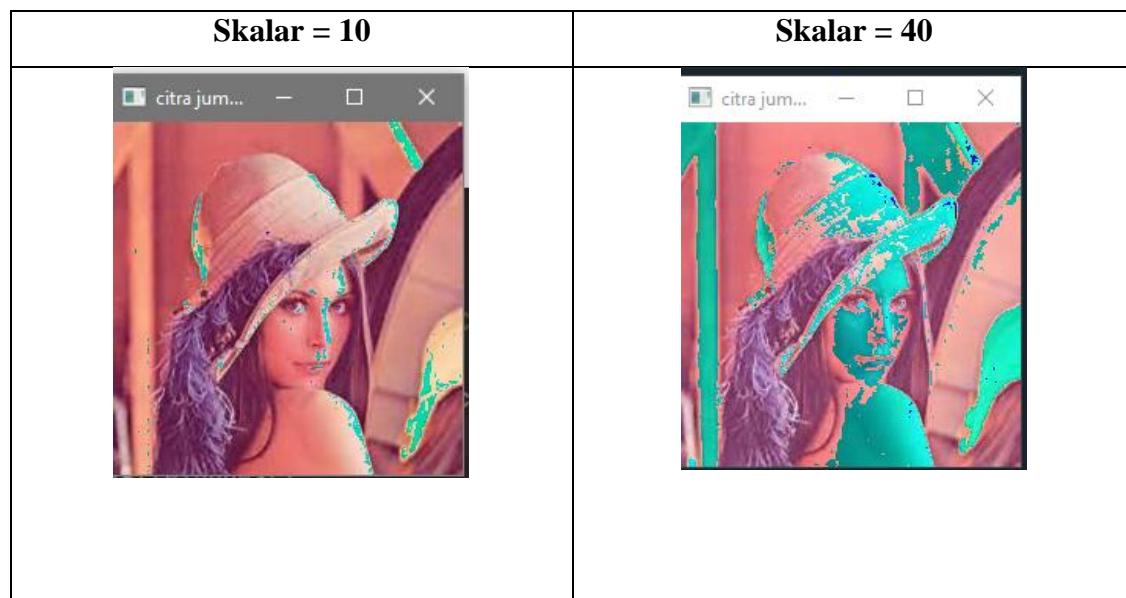
Amati perbedaan dari hasil output tersebut. Jelaskan kenapa hasil output yang didapatkan berbeda!

2. Buatlah program seperti dibawah ini !

```
import cv2

if __name__ == '__main__':
    citra_A = cv2.imread('D:/IMAGE PROCESSING/MATERIAL/lena_warna.jpeg')
    citra_hasil = citra_A + 10
    cv2.imshow('Citra Jumlah Skalar', citra_hasil)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
```

Screenshot hasil output pada program diatas dengan nilai scalar 10, 40, 80, 100





Amati perbedaan dari hasil output tersebut. Jelaskan kenapa hasil output yang didapatkan berbeda!

Matlab
<pre> function fig=JumlahCitraSkalar(A,B) fig=figure; [h w]=size(A); for baris=1:h for kolom=1:w C(baris,kolom)= A(baris,kolom)+ B; end end imshow(C); end </pre>

Pertanyaan :

1. Screenshot hasil output pada program diatas dengan nilai scalar 10, 40, 80, 100

Skalar = 10	Skalar = 40
Skalar = 80	Skalar = 100

Amati perbedaan dari hasil output tersebut. Jelaskan kenapa hasil output yang didapatkan berbeda!

2. Buatlah program seperti dibawah ini !

```
>> A = imread('D:/IMAGE PROCESSING/MATERIAL/lena_warna.jpeg');
>> citra_hasil = A + 10;
>> imshow(citra_hasil);
... |
```

3. Screenshot hasil output pada program diatas dengan nilai scalar 10, 40, 80, 100

Skalar = 10	Skalar = 40
Skalar = 80	Skalar = 100

Amati perbedaan dari hasil output tersebut. Jelaskan kenapa hasil output yang didapatkan berbeda!

5.6 PENGURANGAN CITRA DENGAN SKALAR

Operasi aritmatika pengurangan citra dengan scalar :

$$B(x,y) = A(x,y) - C$$

Python

```
import cv2
import numpy as np

def pengurangan_citra_skalar(citra_A, nilai_skalar):
    A = cv2.imread(citra_A)

    height_A, width_A = A.shape[:2]
    citra_hasil = np.ones([height_A, width_A, 3], np.uint8)
    for x in range(height_A):
        for y in range(width_A):
            citra_hasil[x, y] = A[x, y] - nilai_skalar

    cv2.imshow('Hasil Output Kurang Citra Skalar', citra_hasil)

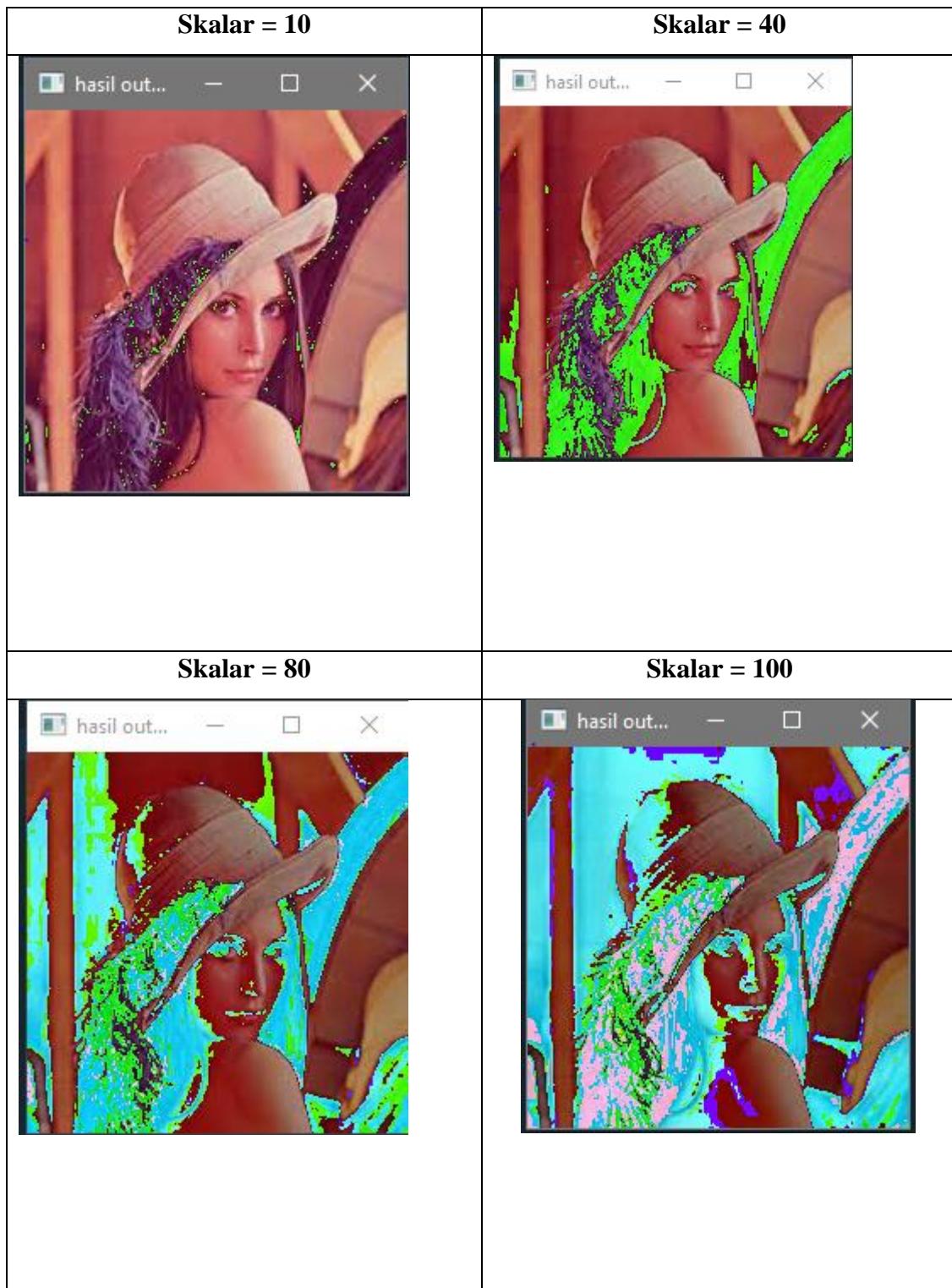
if __name__ == '__main__':
    citra_A = 'D:/IMAGE PROCESSING/MATERIAL/lena_warna.jpeg'

    pengurangan_citra_skalar(citra_A, 10)

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

Pertanyaan :

1. Screenshot hasil output pada program diatas dengan nilai scalar 10, 40, 80, 100



Amati perbedaan dari hasil output tersebut. Jelaskan kenapa hasil output yang didapatkan berbeda!

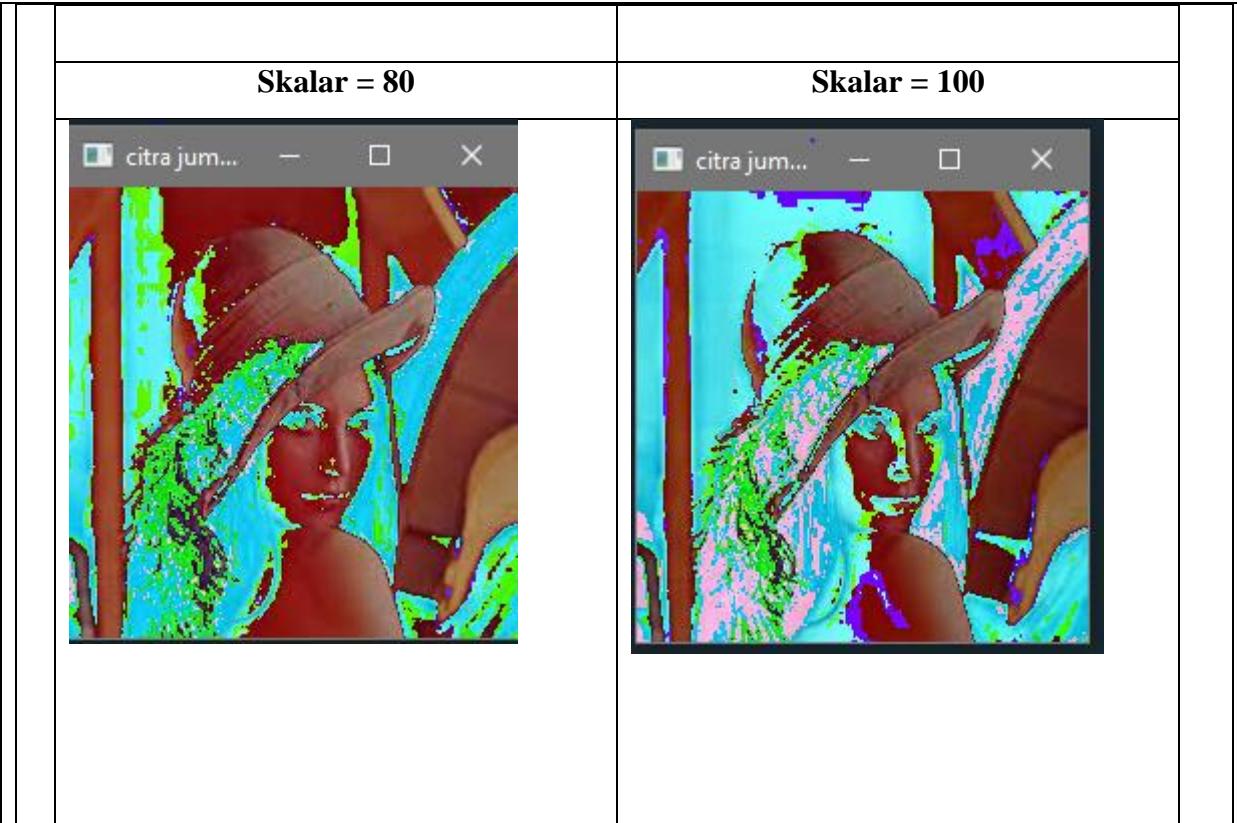
2. Buatlah seperti program di bawah ini

```
import cv2

if __name__ == '__main__':
    citra_A = cv2.imread('D:/IMAGE PROCESSING/MATERIAL/lena_warna.jpeg')
    citra_hasil = citra_A - 10
    cv2.imshow('Citra Jumlah Skalar', citra_hasil)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
```

Screenshot hasil output pada program diatas dengan nilai scalar 10, 40, 80, 100





Amati perbedaan dari hasil output tersebut. Jelaskan kenapa hasil output yang didapatkan berbeda!

Matlab	<pre>function fig=KurangCitraSkalar(A,B) fig=figure; [h w]=size(A); for baris=1:h for kolom=1:w C(baris,kolom)= A(baris,kolom)- B; end end imshow(C); end</pre>
---------------	---

Pertanyaan :

1. Screenshot hasil output pada program diatas dengan nilai scalar 10, 40, 80, 100

Skalar = 10	Skalar = 40
Skalar = 80	Skalar = 100

Amati perbedaan dari hasil output tersebut. Jelaskan kenapa hasil output yang didapatkan berbeda!

2. Buatlah program seperti di bawah ini !

```
>> A = imread('D:/IMAGE PROCESSING/MATERIAL/lena_warna.jpeg');
>> citra_hasil = A - 10;
>> imshow(citra_hasil);
```

Screenshot hasil output pada program diatas dengan nilai scalar 10, 40, 80, 100

Skalar = 10	Skalar = 40
Skalar = 80	Skalar = 100

Amati perbedaan dari hasil output tersebut. Jelaskan kenapa hasil output yang didapatkan berbeda!

5.7 PERKALIAN CITRA DENGAN SKALAR

Operasi aritmatika perkalian citra dengan scalar :

$$B(x,y) = A(x,y) * C$$

Python

Pertanyaan :

- Buat dan jelaskan setiap baris program operasi aritmatik dengan perkalian scalar dengan menggunakan python ! Screenshot program serta hasil output program.

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows a folder named "RASHEED" containing various Python files (divide.py, flipping.py, flowers.jpg, from_PIL_import_Image.py, fruits.jpeg, image_01_gray.jpg, image_02_gray.jpg, lena_grayscale.jpg, lena_warna.jpeg, mawar1.jpg, multiply.py, nomer4_1.py, nomer5_ke2.py, nomer5_pengurangan_citra.py, nomer5.py, nomer6_1.py, operasi_and.py, operasi_bitwiseand.py, operasi_not.py, operasi_or.py, pembagian_citra.py, pembagian.py, pengskalaan_image.py, pengurangan_citra2.py, pengurangan_skalar.py, pengurangan_skalar2.py, penjumlahan_skalar.py, penjumlahan_skalar_bagian3.py, penjumlahan_skalar2.py, perkalian_skalar.py, and uts1(3).py).
- Code Editor:** Displays the content of the "perkalian_skalar.py" file:

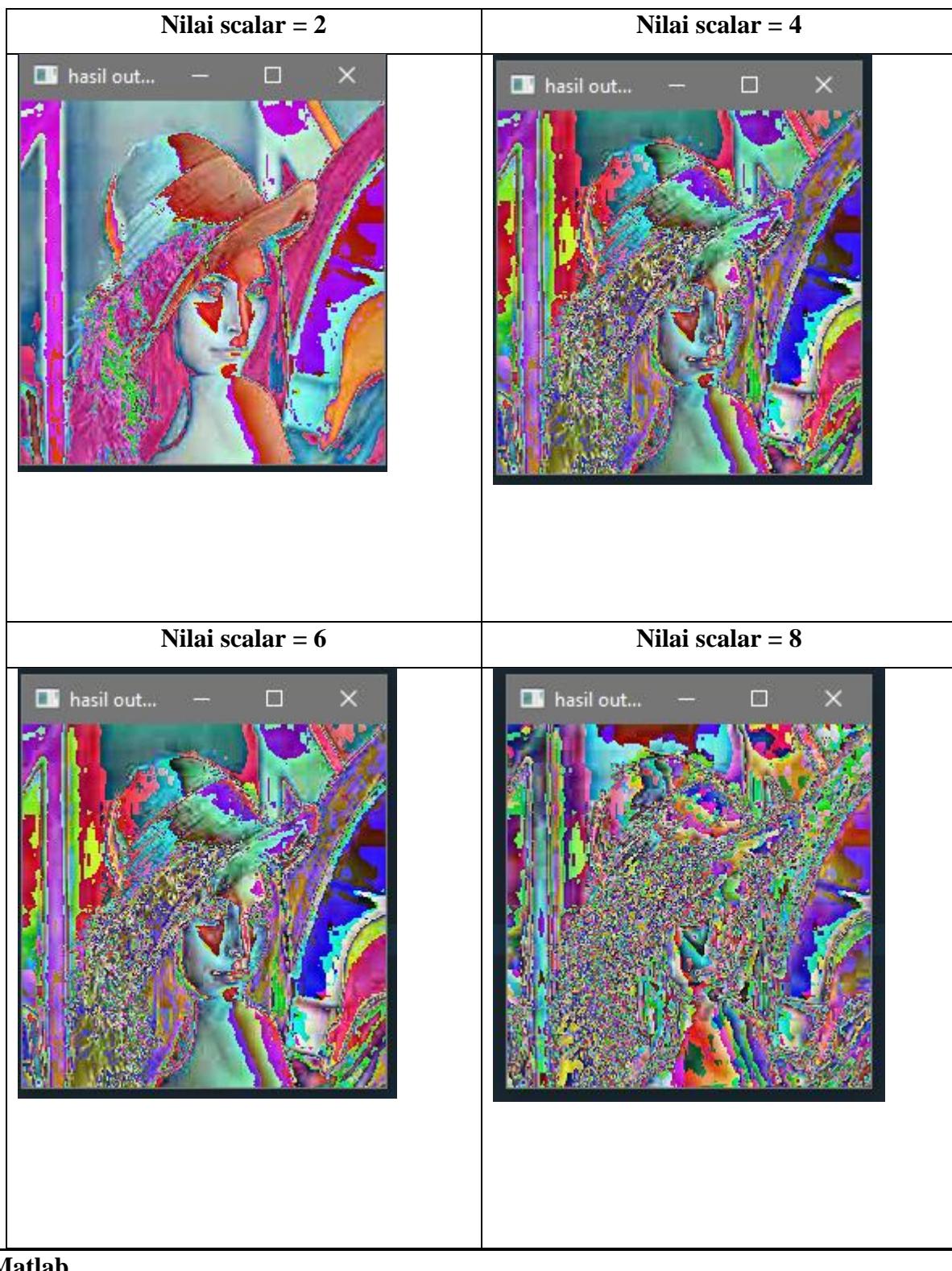
```
import cv2
import numpy as np

def penambahan_citra(citra_A, nilai_skalar):
    A = cv2.imread(citra_A)

    height_A, width_A = A.shape[:2]
    citra_hasil = np.ones([height_A, width_A, 3], np.uint8)
    for x in range(height_A):
        for y in range(width_A):
            citra_hasil[x, y] = A[x, y] * nilai_skalar

    cv2.imshow('hasil output tambah citra', citra_hasil)
if __name__ == '__main__':
    citra_A = 'D:/Rasheed/image_procesing/lena_warna.jpeg'
    penambahan_citra(citra_A, 8)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
```
- Terminal:** Shows the command "perkalian_skalar.py - Rasheed - Visual Studio Code".
- Output:** A small window titled "hasil out..." displays a colorful, distorted image of a flower, which is the result of multiplying the original image by a scalar value of 8.

2. Buatlah dengan nilai scalar antara lain 2, 4, 6, 8 dan screenshot setiap hasilnya serta jelaskan perbedaannya.



Matlab

Pertanyaan :

1. Buat dan jelaskan setiap baris program operasi aritmatik dengan perkalian scalar dengan menggunakan matlab ! Screenshot program serta hasil output program.
2. Buatlah dengan nilai scalar antara lain 2, 4, 6, 8 dan screenshoot setiap hasilnya serta jelaskan perbedaannya.

Nilai scalar = 2	Nilai scalar = 4
Nilai scalar = 6	Nilai scalar = 8

5.8 PEMBAGIAN CITRA DENGAN SKALAR

Operasi aritmatika pembagian citra dengan scalar :

$$B(x,y) = A(x,y) / C$$

Python

Pertanyaan :

- Buat dan jelaskan setiap baris program operasi aritmatik dengan pembagian scalar dengan menggunakan python ! Screenshot program serta hasil output program.

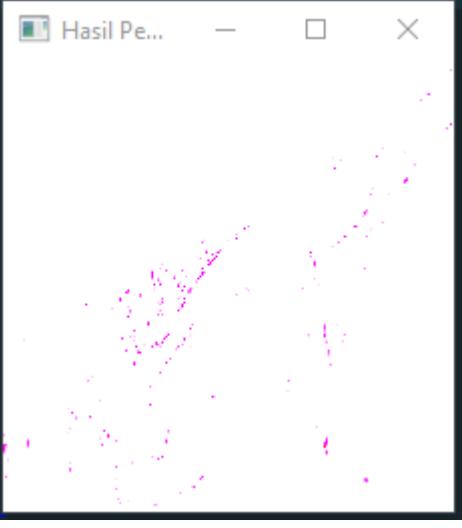
The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left lists various Python files and image files. The main code editor window displays a script named `pembagian_citra.py` with the following content:

```
image proccesing > pembagian_citra.py > ...
1 import cv2 # memasukan Library opencv
2
3 # memasukan citra pada variabel
4 citra = cv2.imread('D:/Rasheed/image proccesing/lena_warna.jpeg')
5
6
7 # operasi pembagian citra dengan skalar
8 citra_hasil = citra / 50
9
10 # menampilkan hasil operasi pembagian citra dengan skalar
11 cv2.imshow('Hasil Pembagian Skalar', citra_hasil)
12 cv2.waitKey(0)
13 cv2.destroyAllWindows()
```

To the right of the code editor is a small window titled "Hasil Pe..." showing a grayscale image of a person's face.

- Buatlah dengan nilai scalar antara lain 2, 4, 6, 8 dan screenshotsetiap hasilnya serta jelaskan perbedaannya.

Nilai scalar = 2	Nilai scalar = 4
A screenshot of a grayscale image titled "Hasil Pe...". The image shows a person's face with a visible vertical color banding effect, indicating the scalar value of 2.	A screenshot of a grayscale image titled "Hasil Pe...". The image shows a person's face with a more pronounced vertical color banding effect compared to the previous one, indicating the scalar value of 4.

Nilai scalar = 6	Nilai scalar = 8
	

Matlab

Pertanyaan :

1. Buat dan jelaskan setiap baris program operasi aritmatik dengan perkalian scalar dengan menggunakan matlab ! Screenshot program serta hasil output program.
2. Buatlah dengan nilai scalar antara lain 2, 4, 6, 8 dan screenshot setiap hasilnya serta jelaskan perbedaannya.

Nilai scalar = 2	Nilai scalar = 4
Nilai scalar = 6	Nilai scalar = 8

6. OPERASI BOOLEAN

6.1 OPERASI AND

Persamaan : $C(x, y) = A(x, y) \text{ and } B(x, y)$

Python

```

import cv2
import numpy as np

def penambahan_citra_and(citra_A, citra_B):
    A = cv2.imread(citra_A)
    B = cv2.imread(citra_B)

    height_A, width_A = A.shape[:2]
    citra_hasil = np.ones([height_A, width_A, 3], np.uint8)
    for x in range(height_A):
        for y in range(width_A):
            citra_hasil[x, y] = A[x, y] & B[x, y]

    cv2.imshow('Hasil Output Tambah Citra', citra_hasil)

if __name__ == '__main__':
    citra_A = 'D:/IMAGE PROCESSING/MATERIAL/mawar1.jpg'
    citra_B = 'D:/IMAGE PROCESSING/MATERIAL/rose2.jpg'

    penambahan_citra_and(citra_A, citra_B)

    cv2.waitKey(0)
    cv2.destroyAllWindows()

```

Pertanyaan :

1. Jelaskan setiap baris koding diatas dan screenshot hasil output program yang dihasilkan dari kode program di atas.

```
import cv2
import numpy as np
```

memasukan library opencv dan numpy:

```
def penambahan_citra(citra_A,citra_B):
    A = cv2.imread(citra_A)
    B = cv2.imread(citra_B)
```

membuat fungsi :

```
height_A, width_A = A.shape[:2]
citra_hasil = np.ones([height_A, width_A, 3], np.uint8)
for x in range (height_A):
    for y in range (width_A):
        citra_hasil[x, y] = A[x, y] & B[x,y]
```

melakukan operasi AND menggunakan pengulangan FOR:

```
cv2.imshow('hasil output tambah citra', citra_hasil)
menampilkan hasil citra
```

melakukan pemanggilan main dengan IF;

```
if __name__ == '__main__':
    citra_A = 'D:/Rasheed/image procesing/mawar1.jpg'
    citra_B = 'D:/Rasheed/image procesing/rose2.jpg'
```

memanggil function:

```
penambahan_citra(citra_A, citra_B)
```

untuk mempertahankan window agar tetap menampilkan gambar:

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

berfungsi untuk menutup dari windows :

```
cv2.destroyAllWindows()
```



2. Buatlah program seperti di bawah ini dengan menggunakan fungsi **cv2.bitwise_and**

```
import cv2

def penambahan_citra_and(citra_A, citra_B):
    A = cv2.imread(citra_A)
    B = cv2.imread(citra_B)

    citra_hasil = cv2.bitwise_and(A, B)

    cv2.imshow('Hasil Output Tambah Citra', citra_hasil)

if __name__ == '__main__':
    citra_A = 'D:/IMAGE PROCESSING/MATERIAL/mawar1.jpg'
    citra_B = 'D:/IMAGE PROCESSING/MATERIAL/rose2.jpg'

    penambahan_citra_and(citra_A, citra_B)

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

Jelaskan setiap baris program diatas dan screenshot hasil output program yang didapatkan.

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows a list of Python files in the current workspace, including `divide.py`, `flipping.py`, `flowers.jpg`, `from_PIL_import...`, `fruits.jpeg`, `image_01_gray.j...`, `image_02_gray.j...`, `lena_grayscale.jpg`, `lena_warmajpeg`, `mawari.jpg`, `multiply.py`, `nomer4_1.py`, `nomer5_ke2.py`, `nomer5_pengur...`, `nomer5.py`, `nomer6_1.py`, `operasi_and.py`, `operasi_bitwisea...`, `operasi_not.py`, `operasi_or.py`, `pembagian_citra...`, `pembagian.py`, `pengkalaan_im...`, `pengurangan_ci...`, `pengurangan_sk...`, `pengurangan_sk...`, `penjumlahan_sa...`, `penjumlahan_sk...`, `penjumlahlahcitra...`, `perkalian_skalar...`, `perkalian_skalar...`, `perkalian.py`, `perkalian2.py`, `resize.py`, `resize2.5.jpg`, `RGB_Citra.png`, `rose2.jpg`, `rotat`, `spongboob1.JPG`, `translasi.py`, `uts1(2).py`, `uts1(3).py`, and `warpaffine.py`.
- Code Editor:** Displays the `operasi_bitwiseand.py` file with the following content:

```
import cv2

def penambahan_citra_and(citraA, citraB):
    A = cv2.imread(citraA)
    B = cv2.imread(citraB)

    citra_hasil = cv2.bitwise_and(A, B)

    cv2.imshow('Hasil Output operasi BitwiseAND', citra_hasil)

if __name__ == '__main__':
    citraA = 'D:/Rasheed/image proccesing/mawari1.jpg'
    citraB = 'D:/Rasheed/image proccesing/rose2.jpg'

    penambahan_citra_and(citraA, citraB)

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

- Terminal:** Shows the command `cv2.destroyAllWindows()` and several `AttributeError` messages from previous runs.
- Output:** Shows the output window titled "Hasil Output operasi BitwiseAND" displaying a black silhouette of two roses on a background of white roses.
- Bottom Status Bar:** Shows the path "Rasheed/image proccesing/multiply.py" and other file paths.

Amati dan jelaskan perbedaan yang didapatkan dari program sebelumnya ! jelaskan alasan yang membedakannya.

Matlab

```

function fig=CitraAnd(A,B)
    fig=figure;
    [h w]=size(A);
    for baris=1:h
        for kolom=1:w
            C(baris,kolom)= and(A(baris, kolom), B(baris,kolom));
        end
    end
    imshow(C);
end

```

Pertanyaan :

1. Jelaskan setiap baris koding diatas dan screenshot program untuk menjalankan fungsi tersebut serta hasil output program yang dihasilkan dari kode program di atas.
2. Buatlah program seperti dibawah ini dengan menggunakan **bitand()**

```
>> A = imread('D:/IMAGE PROCESSING/MATERIAL/mawar1.jpg');
>> B = imread('D:/IMAGE PROCESSING/MATERIAL/rose2.jpg');
>> citra_hasil = bitand(A,B);
>> imshow(citra_hasil)
```

Jelaskan setiap baris program diatas dan screenshot hasil output program yang didapatkan.

Amati dan jelaskan perbedaan yang didapatkan dari program sebelumnya ! jelaskan alasan yang membedakannya.

6.2 OPERASI OR

Persamaan :

$$C(x, y) = A(x, y) \text{ or } B(x, y)$$

Python

Pertanyaan :

1. Buatlah program operasi Boolean dengan menggunakan operator OR pada 2 buah citra dan jelaskan setiap baris program yang dibuat serta screenshot program yang dibuat dan hasil output program yang dibangun

```
import cv2
```

berfungsi untuk memanggil package atau modul cv2

```
citraA = cv2.imread('D:/Rasheed/image proccesing/mawar1.jpg')
citraB = cv2.imread('D:/Rasheed/image proccesing/rose2.jpg')
```

berfungsi untuk memanggil gambar dari dalam folder kita ke variable

```
citra_hasil = citraA / citraB
```

operasi OR antara citra

```
cv2.imshow('Hasil operasi OR', citra_hasil)
```

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

menampilkan hasil citra dari pada operasi OR citra yang di lakukan di atas



2. Buatlah program operasi citra dengan menggunakan operasi OR pada fungsi **cv2.bitwise_or()** pada 2 buah citra dan jelaskan setiap baris program yang dibuat serta screenshot program yang dibuat dan hasil output program yang dibangun.

```
import cv2
```

berfungsi untuk memanggil package atau modul cv2

```
citraA = cv2.imread('D:/Rasheed/image proccesing/mawar1.jpg')
```

```
citraB = cv2.imread('D:/Rasheed/image proccesing/rose2.jpg')
```

berfungsi untuk memanggil gambar dari dalam folder kita ke variable

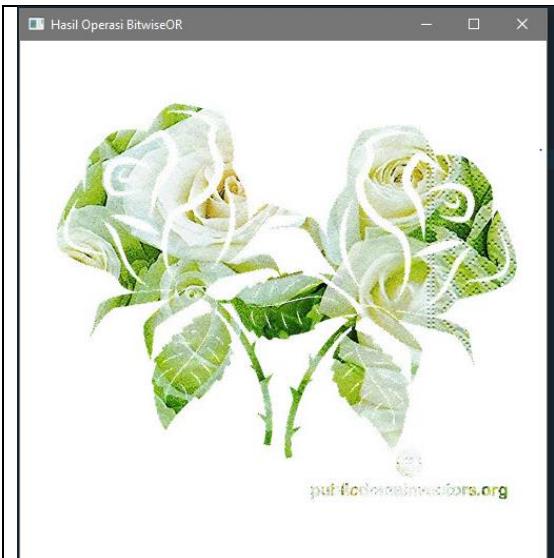
```
citra_hasil = cv2.bitwise_or(citra_A,citra_B)
```

fungsi **cv2.bitwise_or()** pada 2 buah citra

```
cv2.imshow('Hasil operasi OR', citra_hasil)
```

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

menampilkan hasil citra dari pada operasi OR citra yang di lakukan di atas



Matlab

Pertanyaan :

- 1 Buatlah program operasi Boolean dengan menggunakan operator OR pada 2 buah citra dan jelaskan setiap baris program yang dibuat serta screenshot program yang dibuat dan hasil output program yang dibangun.
- 2 Buatlah program operasi citra dengan menggunakan operasi OR pada fungsi **bitor()** pada 2 buah citra dan jelaskan setiap baris program yang dibuat serta screenshot program yang dibuat dan hasil output program yang dibangun.

6.3 OPERASI NOT

Persamaan :

$$C(x, y) = \text{not } A(x, y)$$

Python

Pertanyaan :

1. Buatlah program operasi Boolean dengan menggunakan operator NOT pada citra dan jelaskan setiap baris program yang dibuat serta screenshot program yang dibuat dan hasil output program yang dibangun.

```
import cv2
```

berfungsi untuk memanggil package atau modul cv2

```
citraA = cv2.imread('D:/Rasheed/image proccesing/mawar1.jpg')  
citraB = cv2.imread('D:/Rasheed/image proccesing/rose2.jpg')
```

berfungsi untuk memanggil gambar dari dalam folder kita ke variable

```
citra_hasil = ~ citra_A
```

operasi OR antara 2 buah citra

```
cv2.imshow('Hasil operasi not', citra_hasil)  
cv2.waitKey(0)
```

menampilkan hasil citra dari pada operasi NOT citra yang di lakukan di atas



2. Buatlah program operasi citra dengan menggunakan operasi NOT pada fungsi `cv2.bitwise_not()` pada citra dan jelaskan setiap baris program yang dibuat serta screenshot program yang dibuat dan hasil output program yang dibangun.

```
import cv2
```

berfungsi untuk memanggil package atau modul cv2

```
citraA = cv2.imread('D:/Rasheed/image proccesing/mawar1.jpg')  
citraB = cv2.imread('D:/Rasheed/image proccesing/rose2.jpg')
```

berfungsi untuk memanggil gambar dari dalam folder kita ke variable

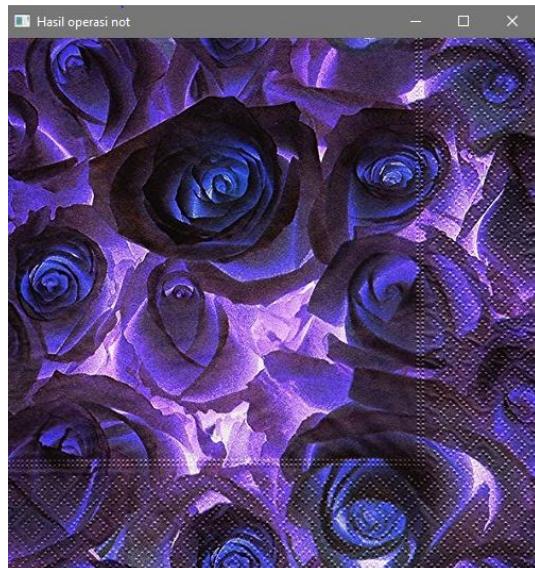
```
citra_hasil = cv2.bitwise_not(citra_A,citra_B)
```

fungsi `cv2.bitwise_not()` pada 2 buah citra

```
cv2.imshow('Hasil operasi not', citra_hasil)
```

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

menampilkan hasil citra dari pada operasi OR citra yang di lakukan di atas



Matlab

Pertanyaan :

1. Buatlah program operasi Boolean dengan menggunakan operator NOT pada 2 buah citra dan jelaskan setiap baris program yang dibuat serta screenshot program yang dibuat dan hasil output program yang dibangun.

2. Buatlah program operasi citra dengan menggunakan operasi NOT pada fungsi **objout()** pada 2 buah citra dan jelaskan setiap baris program yang dibuat serta screenshot program yang dibuat dan hasil output program yang dibangun.

6.4 OPERASI XOR

Persamaan :

$$C(x, y) = A(x, y) \text{ xor } B(x, y)$$

Python

Pertanyaan :

1. Buatlah program operasi Boolean dengan menggunakan operator XOR pada citra dan jelaskan setiap baris program yang dibuat serta screenshot program yang dibuat dan hasil output program yang dibangun.

```
import cv2
```

berfungsi untuk memanggil package atau modul cv2

```
citraA = cv2.imread('D:/Rasheed/image proccesing/mawar1.jpg')
citraB = cv2.imread('D:/Rasheed/image proccesing/rose2.jpg')
```

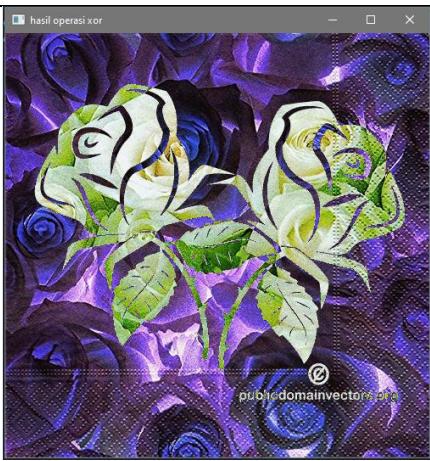
berfungsi untuk memanggil gambar dari dalam folder kita ke variable

```
citra_hasil = citra_A ^ citra_B
```

operasi XOR antara 2 buah citra

```
cv2.imshow('hasil operasi xor ', citra_hasil)
cv2.waitKey(0)
```

menampilkan hasil citra dari pada operasi XOR citra yang di lakukan di atas



2. Buatlah program operasi citra dengan menggunakan operasi XOR pada fungsi `cv2.bitwise_xor()` pada citra dan jelaskan setiap baris program yang dibuat serta screenshot program yang dibuat dan hasil output program yang dibangun

```
. import cv2
```

berfungsi untuk memanggil package atau modul cv2

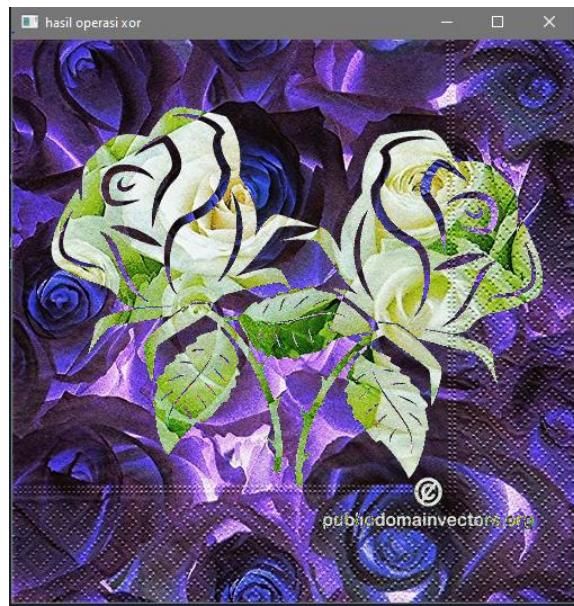
```
citraA = cv2.imread('D:/Rasheed/image proccesing/mawar1.jpg')  
citraB = cv2.imread('D:/Rasheed/image proccesing/rose2.jpg')
```

berfungsi untuk memanggil gambar dari dalam folder kita ke variable

```
citra_hasil = cv2.bitwise_xor(citra_A,citra_B)  
fungsi cv2.bitwise_xor() pada 2 buah citra
```

```
cv2.imshow('Hasil operasi xor', citra_hasil)  
cv2.waitKey(0)
```

menampilkan hasil citra dari pada operasi XOR citra yang di lakukan di atas



Matlab

Pertanyaan :

1. Buatlah program operasi Boolean dengan menggunakan operasi XOR pada citra dan jelaskan setiap baris program yang dibuat serta screenshot program yang dibuat dan hasil output program yang dibangun.
2. Buatlah program operasi citra dengan menggunakan operasi XOR pada fungsi **bitxor()** pada citra dan jelaskan setiap baris program yang dibuat serta screenshot program yang dibuat dan hasil output program yang dibangun.

3. OPERASI GEOMETRI

1. TRANSLASI

Persamaan :

$$x' = x + m$$

$$y' = y + n$$

Dan

$$B[x][y] = A[x + m][y + n]$$

Python

```
1 import cv2
2 import numpy as np
3
4 def translasi(m, n):
5     citra = cv2.imread('D:/IMAGE PROCESSING/MATERIAL/fruits.jpeg')
6
7     height, width = citra.shape[:2]
8     citra_hasil = np.ones([height, width, 3], np.uint8)
9
10    start_m = m
11    start_n = n
12
13    if m < 0:
14        start_m = 0
15    if n < 0:
16        start_n = 0
17
18    for x in range(start_m, height):
19        for y in range(start_n, width):
20            x_new = x - m
21            y_new = y - n
22
23            if(x_new >= height or y_new >= width or x_new < 0 or y_new < 0):
24                citra_hasil[x, y] = (0, 0, 0)
25            else:
26                citra_hasil[x, y] = citra[x, y]
27
28    cv2.imshow('Hasil Translasi', citra_hasil)
```

Pertanyaan :

1. Silakan screen shoot setiap di program main pada Python untuk memanggil fungsi translasi dengan nilainya adalah
 - a. Nilai translasi pada titik x dan y adalah 100 dan 400

```
import cv2
import numpy as np
def translasi(m, n):
    citra = cv2.imread('D:/Rasheed/image proccesing/fruits.jpeg')
    (H, W) = citra.shape[:2]
    citra_hasil = np.ones([H, W, 3], np.uint8)
    start_m = m;
    start_n = n;
    if m < 0:
        start_m = 0
    if n < 0:
        start_n = 0
    for x in range(start_m, H):
        for y in range(start_n, W):
            x_new = x - m
            y_new = y - n
            if(x_new >= H or y_new >= W or x_new < 0 or y_new < 0):
                citra_hasil[x, y] = (0, 0, 0)
            else:
                citra_hasil[x, y] = citra[x, y]
    cv2.imshow('Hasil Translasi', citra_hasil)
if __name__ == '__main__':
    translasi(-100,-400)
cv2.waitKey(0)
```

- b. Nilai translasi pada titik x dan y adalah -100 dan -400

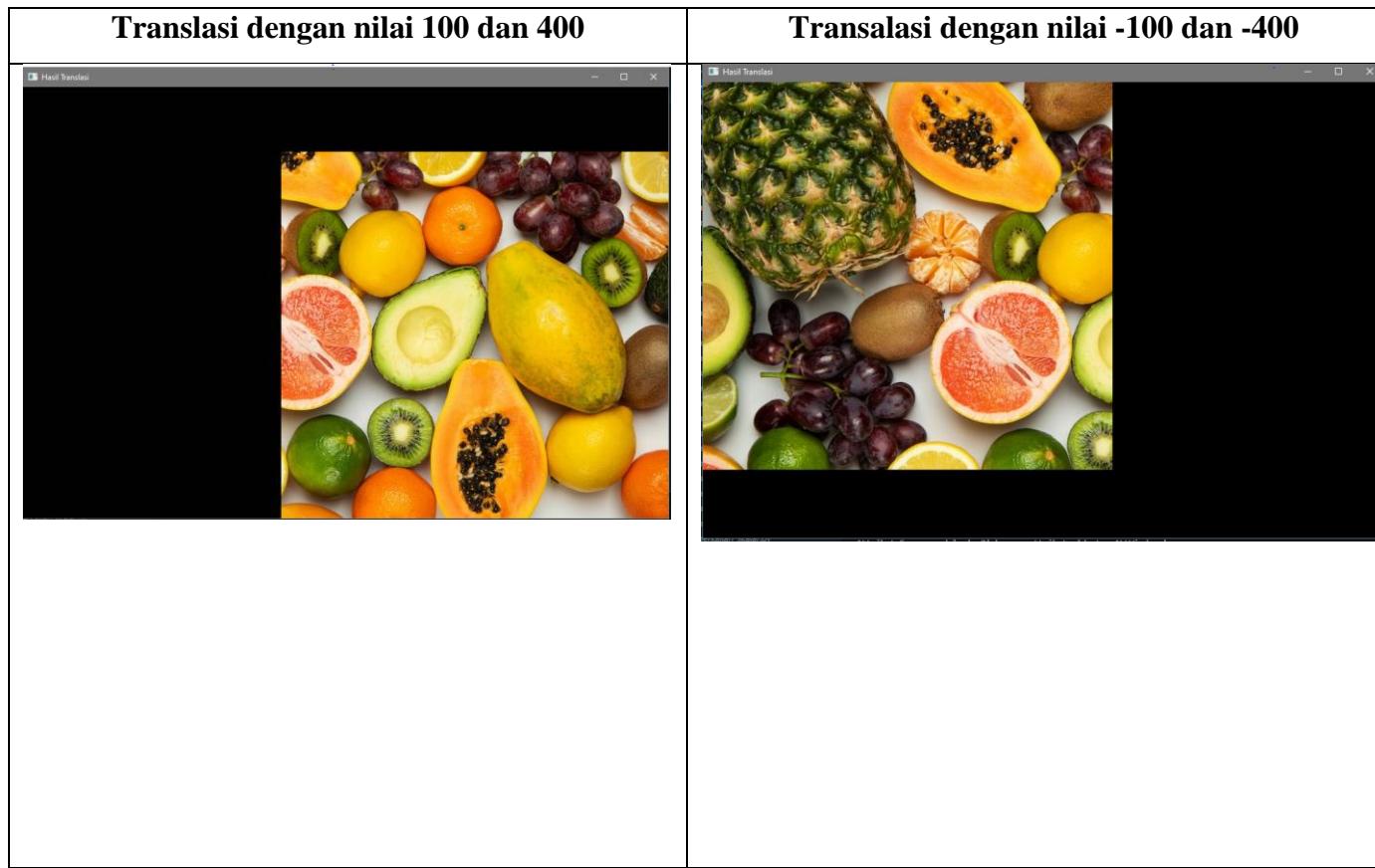
```
import cv2
import numpy as np
def translasi(m, n):
    citra = cv2.imread('D:/Rasheed/image proccesing/fruits.jpeg')
    (H, W) = citra.shape[:2]
    citra_hasil = np.ones([H, W, 3], np.uint8)
    start_m = m;
    start_n = n;
    if m < 0:
        start_m = 0
    if n < 0:
```

```

start_n = 0
for x in range(start_m, H):
    for y in range(start_n, W):
        x_new = x - m
        y_new = y - n
        if(x_new >= H or y_new >= W or x_new < 0 or y_new < 0):
            citra_hasil[x, y] = (0, 0, 0)
        else:
            citra_hasil[x, y] = citra[x, y]
cv2.imshow('Hasil Translasi', citra_hasil)
if __name__ == '__main__':
    translasi(-100,-400)
cv2.waitKey(0)

```

2. Screen shot hasil traslasi sesuai dengan soal 1 sebelumnya untuk pembahasan translasi.



3. Buat pemrograman translasi citra dengan menggunakan fungsi **cv2.warpAffine()** pada Python dengan nilai x dan y sama seperti pada soal no.1 pada sub-bab translasi diatas. (Screenshot program dan hasil eksekusi program) !\

```

import cv2
import numpy as np

def translasi_citra(x, y):

    citra = cv2.imread('D:/Rasheed/image proccesing/fruits.jpeg')
    rows,cols = citra.shape[:2]

    M = np.float32([[1,0,x],[0,1,y]])
    citra_hasil = cv2.warpAffine(citra,M,(cols,rows))

    cv2.imshow('Hasil Translasi Citra', citra_hasil)

if __name__ == '__main__':
    translasi_citra(100, 400)

cv2.waitKey(0)
cv2.destroyAllWindows()

```



x = 100 y = 400



X = -100 y = -400

Matlab

```
1 function fig=TranslasiCitra(A, M, N)
2 -
3 -     fig=figure;
4 -     [height width] = size(A);
5 -     for x=1:height
6 -         for y=1:width
7 -             citra_hasil(x+M, y+N) = A(x, y);
8 -         end
9 -     end
10 -    imshow(citra_hasil);
11 end
```

Pertanyaan :

1. Silakan screen shoot setiap di program main pada Matlab untuk memanggil fungsi translasi dengan nilainya adalah
 - a. Nilai transalasi pada titik x dan y adalah 100 dan 400
 - b. NIlai translasi pada titik x dan y adalah -100 dan -400
 - c. Screen shot hasil traslasi sesuai dengan soal 1 sebelumnya untuk pembahasan translasi.

Translasi dengan nilai 100 dan 400

Transalasi dengan nilai -100 dan -400

- | | | |
|--|--|--|
| | | |
|--|--|--|
- d. Buat pemrograman translasi citra dengan menggunakan fungsi **imtranslate** pada Matlab dengan nilai parameter translasi sama seperti soal no.1 pada sub-bab translasi .
(Screenshot program dan hasil eksekusi program) !

2. ROTASI

Dengan persamaan dengan sudut berlawanan arah jarum jam :

$$B[x'][y'] = B[x \cos(\theta) - y \sin(\theta)][x \cos(\theta) + y \sin(\theta)] = A[x][y]$$

Python

```
# Referensi : https://github.com/imambungo/operasi-citra
from PIL import Image
from math import sin, cos

def rotasi(derajat):
    citra = Image.open('D:/IMAGE PROCESSING/MATERIAL/spongboob1.jpg')
    citra_pixel = citra.load()

    height = citra.size[0]
    width = citra.size[1]

    citra_baru = Image.new("RGB", (height, width))
    citra_baru_pixel = citra_baru.load()

    x_tengah = height // 2
    y_tengah = width // 2

    for x in range(height):
        for y in range(width):
            theta = derajat * 3.14 / 180

            x_baru = cos(theta) * (x-x_tengah) - sin(theta) * (y-y_tengah) + x_tengah
            y_baru = sin(theta) * (x-x_tengah) + cos(theta) * (y-y_tengah) + y_tengah
            #x_baru = x*cos(theta) - y*sin(theta)
            #y_baru = x*cos(theta) + y*sin(theta)

            if(x_baru >= height or y_baru >= width or x_baru < 0 or y_baru < 0):
                citra_baru_pixel[x, y] = (0, 0, 0)
            else:
                citra_baru_pixel[x, y] = citra_pixel[x_baru, y_baru]

    citra_baru.save('D:/IMAGE PROCESSING/MATERIAL/hasil_rotasi.jpg')

if __name__ == '__main__':
    rotasi(45)
```

Pertanyaan :

1. Buatlah program untuk mengeksekusi fungsi dengan derajat 45 dan -45 dan screenshot hasil tersebut.

Rotasi 45 derajat	Rotasi dengan -45 derajat
	

2. Buatlah dan screenshot program dan hasil eksekusi di matlab dengan menggunakan fungsi **rotate()** dengan derajat 45 dan -45.

Rotasi 45 derajat	Rotasi dengan -45 derajat

Matlab

Membuat fungsi rotasi di matlab dengan kode program sebagai berikut

```

1 - [-] function J = rotasi(I,T)
2 -     m = size(I,1);
3 -     n = size(I,2);
4 -     if rem(m,2) == 0
5 -         Xp = floor((m+1)/2)+1;
6 -     else Xp = floor((m+1)/2);
7 -     end
8 -     if rem(n,2) == 0 Yp = floor((n+1)/2)+1;
9 -     else Yp = floor((n+1)/2);
10 -    end
11 -    X = zeros(m,n);
12 -    Y = zeros(m,n);
13 -    for x = 1 : m
14 -        X(x,1:n) = x;
15 -    end
16 -    for y = 1 : n
17 -        Y(1:m,y) = y;
18 -    end
19 -    Xa = round(Xp + (X - Xp)*cosd(T) - (Y - Yp)*sind(T));
20 -    Ya = round(Yp + (X - Xp)*sind(T) + (Y - Yp)*cosd(T));
21 -    r = size(min(Xa(:)) : max(Xa(:)),2);
22 -    c = size(min(Ya(:)) : max(Ya(:)),2);
23 -    xs = round(abs(r-m)/2);
24 -    ys = round(abs(c-n)/2);
25 -    J = zeros(r,c);
26 -    for x = 1 : m
27 -        for y = 1 : n
28 -            J(Xa(x,y)+xs,Ya(x,y)+ys) = I(x,y);
29 -        end
30 -    end
31 - end

```

Program untuk menjalankan eksekusi fungsi rotasi dengan derajat 45.

```

>> A = imread('D:/IMAGE PROCESSING/MATERIAL/spongboobl.jpg');
>> J = rotasi(A, 45);
>> imshow(uint8(J))

```

Pertanyaan :

1. Buatlah program untuk mengeksekusi fungsi dengan derajat 45 dan -45 dan screenshot hasil tersebut.

Rotasi 45 derajat	Rotasi dengan -45 derajat

2. Buatlah dan screenshot program dan hasil eksekusi di matlab dengan menggunakan fungsi **imrotate()** dengan derajat 45 dan -45.

Rotasi 45 derajat	Rotasi dengan -45 derajat

3. FLIPPING

Flipping horizontal dengan persamaan :

$$B[x][y] = A[N - x][y]$$

Python

Flipping/Pencerminan citra secara horizontal

```
import cv2
import numpy as np

def flip_horizontal():
    citra = cv2.imread('D:/IMAGE PROCESSING/MATERIAL/spongboob1.jpg')

    (height, width) = citra.shape[:2]
    citra_hasil = np.ones([height, width, 3], np.uint8)

    for x in range(height):
        for y in range(width):
            citra_hasil[x, y] = citra[x, width - 1 - y]

    cv2.imshow('Citra hasil flip horizontal', citra_hasil)

if __name__ == '__main__':
    flip_horizontal()

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

Hasil Output yang didapat pada saat di eksekusi :



Pertanyaan :

1. Silakan buat fungsi di python untuk melakukan pencerminan secara vertical, pencerminan pada titik asal. (Screenshot kode program maupun hasil eksekusi).

Flipping Vertikal

Dengan persamaan :

$$B[x][y] = A[x][M - y]$$

```
import cv2
import numpy as np

citra = cv2.imread('D:/Rasheed/image proccesing/spongboob1.jpg')
(H, W) = citra.shape[:2]

citra_hasil = np.ones([H, W, 3], np.uint8)

for x in range(H):
    for y in range(W):
        citra_hasil[x, y] = citra[H - 1 - x, y]

cv2.imshow('Citra Hasil Flip', citra_hasil)

cv2.waitKey(0)
```

**Flipping Titik Asal**

Dengan persamaan :

$$B[x][y] = A[N - x][M - y]$$

```

import cv2
import numpy as np

citra = cv2.imread('D:/Rasheed/image proccesing/spongeboob1.jpg')
(H, W) = citra.shape[:2]

citra_hasil = np.ones([H, W, 3], np.uint8)

for x in range(H):
    for y in range(W):
        citra_hasil[x, y] = citra[H - 1 - x, W - 1 - y]

cv2.imshow('Citra Hasil Flip', citra_hasil)

cv2.waitKey(0)

```



2. Buatlah program untuk melakukan pencerminan secara horizontal, vertical, maupun titik asal dengan menggunakan fungsi **cv2.flip** pada python. (Screenshot kode program maupun hasil eksekusi)

Horizontal

```

import cv2

citra = cv2.imread('D:/Rasheed/image proccesing/spongeboob1.jpg')

citra_hasil = cv2.flip(citra, 1)

cv2.imshow('Hasil Flipping', citra_hasil)
cv2.waitKey(0)

```



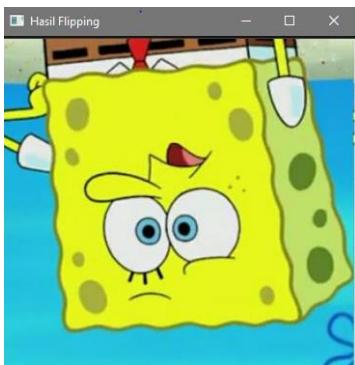
Vertical

```
import cv2

citra = cv2.imread('D:/Rasheed/image proccesing/spongboob1.jpg')

citra_hasil = cv2.flip(citra, 0)

cv2.imshow('Hasil Flipping', citra_hasil)
cv2.waitKey(0)
```



Titik asal

```
import cv2

citra = cv2.imread('D:/Rasheed/image proccesing/spongboob1.jpg')

citra_hasil = cv2.flip(citra, -1)

cv2.imshow('Hasil Flipping', citra_hasil)
cv2.waitKey(0)
```



Matlab

```
function fig = flipHorizontal(A)
    fig=figure;
    [baris kolom] = size(A);
    for x=1:baris
        for y=1:kolom
            citra_hasil(x, (kolom +1)-y) = A(x,y);
        end
    end
    imshow(citra_hasil);
end
```

Hasil Output yang didapat pada saat di eksekusi :



Pertanyaan :

1. Silakan buat fungsi di python untuk melakukan pencerminan secara vertical , pencerminan pada titik asal. (Screenshot kode program maupun hasil eksekusi)

Flipping Vertikal

Dengan persamaan : $B[x][y] = A[x][M - y]$

Flipping Titik Asal

Dengan persamaan : $B[x][y] = A[N - x][M - y]$

2. Buatlah program untuk melakukan pencerminan secara horizontal, vertical, maupun titik asal dengan menggunakan fungsi **flip()** pada matlab. (Screenshot kode program maupun hasil eksekusi).

4. PENSKALAAN IMAGE

Dengan persamaan : $x' = s_x \cdot x$

$$y' = s_y \cdot y$$

Sehingga, menjadi persamaan : $B[x'][y'] = A[s_x \cdot x][s_y \cdot y]$

Python

```
# Reference : https://github.com/imambungo/operasi-citra
from PIL import Image
from math import floor

def skala(nilai):
    citra = Image.open('D:/IMAGE PROCESSING/MATERIAL/spongebob1.jpg')
    citra_pixel = citra.load()

    height = citra.size[0]
    width = citra.size[1]

    height_new = floor(height * 2)
    width_new = floor(width * 2)

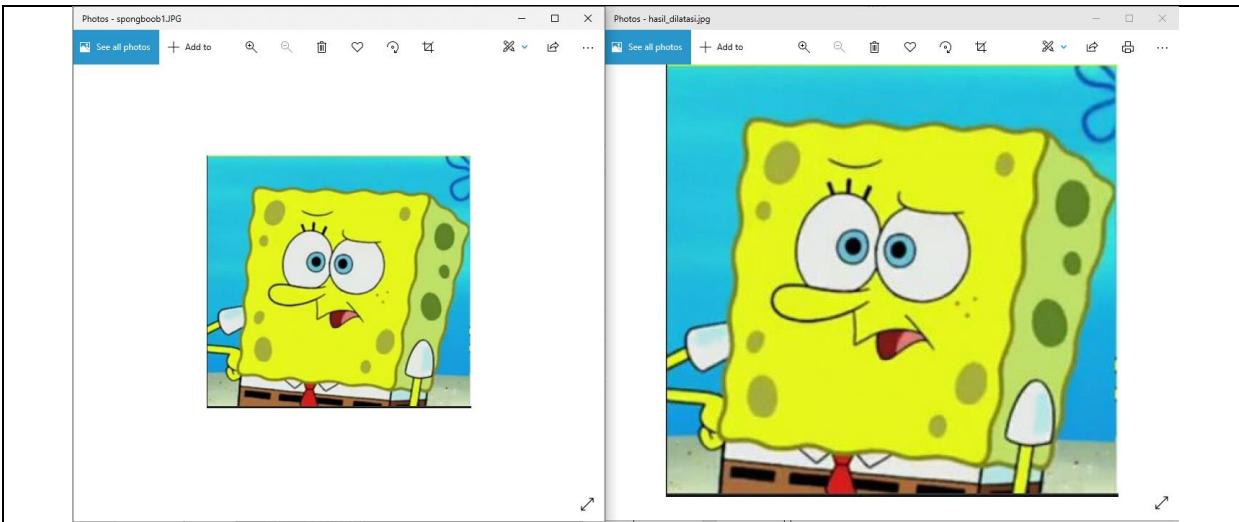
    citra_hasil = Image.new("RGB", (height_new, width_new))
    citra_hasil_pixel = citra_hasil.load()

    for x in range(height_new):
        for y in range(width_new):
            x_ = height * x / height_new
            y_ = width * y / width_new
            citra_hasil_pixel[x,y] = citra_pixel[x_, y_]

    citra_hasil.save('D:/IMAGE PROCESSING/MATERIAL/hasil_dilatasi.jpg')

if __name__ == '__main__':
    skala(0.7)
```

Hasil Output yang didapat pada saat di eksekusi dengan skala perbesaran 0.7 :



Pertanyaan :

1. Buatlah baris kode untuk menampilkan perbesaran citra dengan skala 1.8 dan 2.5 (Screenshot kode program dan hasil perbesaran citra tersebut dengan skala nilai tersebut)

```
from PIL import Image
from math import floor

def skala(nilai):
    citra = Image.open('bahan\\spongboob1.jpg')
    citra_pixel = citra.load()

    H = citra.size[0]
    W = citra.size[1]

    hnew = floor(H * nilai)
    wnew = floor(W * nilai)

    citra_hasil = Image.new("RGB", (hnew, wnew))
    citra_hasil_pixel = citra_hasil.load()

    for x in range(hnew):
        for y in range(wnew):
            x_ = H * x / hnew
            y_ = W * y / wnew
            citra_hasil_pixel[x, y] = citra_pixel[x_, y_]

    citra_hasil.save('geometri\\hasilresize2.5.jpg')
    #citra_hasil.show()
```

```

if __name__ == '__main__':
    skala(2.5)

```

Skala 1.8	Skala 2.5
  1.8 2.5  	  1.8 2.5  

2. Buatlah baris kode untuk menampilkan perbesaran citra dengan skala 1.8 dan 2.5 dengan menggunakan fungsi **resize()** pada python (screenshot kode program dan hasil perbesaran citra tersebut dengan skala nilai tersebut)

Skala 1.8

```

import cv2

citra = cv2.imread('D:/Rasheed/image proccesing/spongboob1.jpg')
skala = 1.8

width = int(citra.shape[1] * skala)
height = int(citra.shape[0] * skala)

shape = (width, height)

```

```

hasil = cv2.resize(citra, shape)

cv2.imshow('resize1.8.jpg', hasil)
#cv2.imshow('Hasil Resize', hasil)
cv2.waitKey(0)

```

skala 2.5

```

import cv2

citra = cv2.imread('D:/Rasheed/image proccesing/spongboob1.jpg')
skala = 2.5

width = int(citra.shape[1] * skala)
height = int(citra.shape[0] * skala)

shape = (width, height)

hasil = cv2.resize(citra, shape)

cv2.imshow('resize2.5.jpg', hasil)
#cv2.imshow('Hasil Resize', hasil)
cv2.waitKey(0)

```

Skala 1.8



Skala 2.5

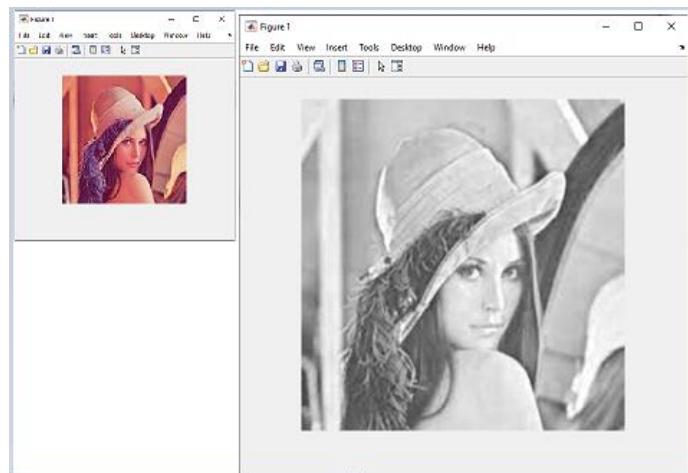


Matlab

```
function J = perbesar(I, ShX, ShY)
    m = size(I,1);
    n = size(I,2);
    r = m*ShX;
    c = n*ShY;
    J = zeros(r,c);
    for x = 1 : m
        for y = 1 : n
            J((x-1)*ShX+1 : x*ShX, (y-1)*ShY+1 : y*ShY) = I(x,y);
        end
    end
end
```

Hasil Output yang didapat pada saat di eksekusi dengan skala perbesaran dengan titik x=2 dan titik y=2 :

```
>> A = imread('D:/IMAGE PROCESSING/MATERIAL/lena_warna.jpeg');
>> B = 2;
>> C = 2;
>> figure, imshow(A);
>> J = perbesar(A, B, C);
>> imshow(uint8(J));
```



Pertanyaan :

1. Buatlah baris kode untuk menampilkan perbesaran citra dengan skala ($x=1, y=2$) dan skala ($x=2, y=3$). (screenshot kode program dan hasil perbesaran citra tersebut dengan skala nilai tersebut)

Skala (x=1, y=2)	Skala (x=2, y=3)

2. Buatlah baris kode untuk menampilkan perbesaran citra dengan skala 1.8 dan 2.5 dengan menggunakan fungsi **imresize()** pada matlab (screenshot kode program dan hasil perbesaran citra tersebut dengan skala nilai tersebut)

Skala 1.8	Skala 2.5

