Pengolahan Citra

Install python, IDE, opencv, menampilkan gambar

Dosen Pengampu

Hero Yudo Martono ST, MT

Disusun Oleh :

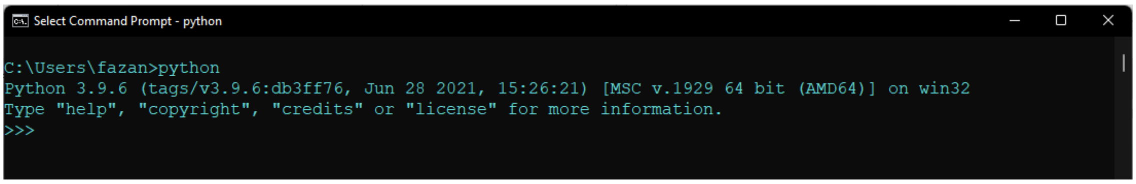
Nama	:	M. Faza Nur Husain
Nrp	:	3121550004

D3 PJJ AK TEKNIK INFORMATIKA

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

TAHUN AKADEMIK 2021/2022

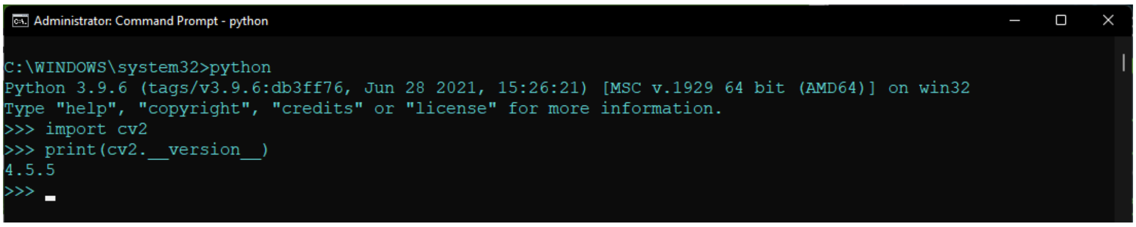
1. Install python



```
C:\Users\fazan>python
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

2. Install text editor

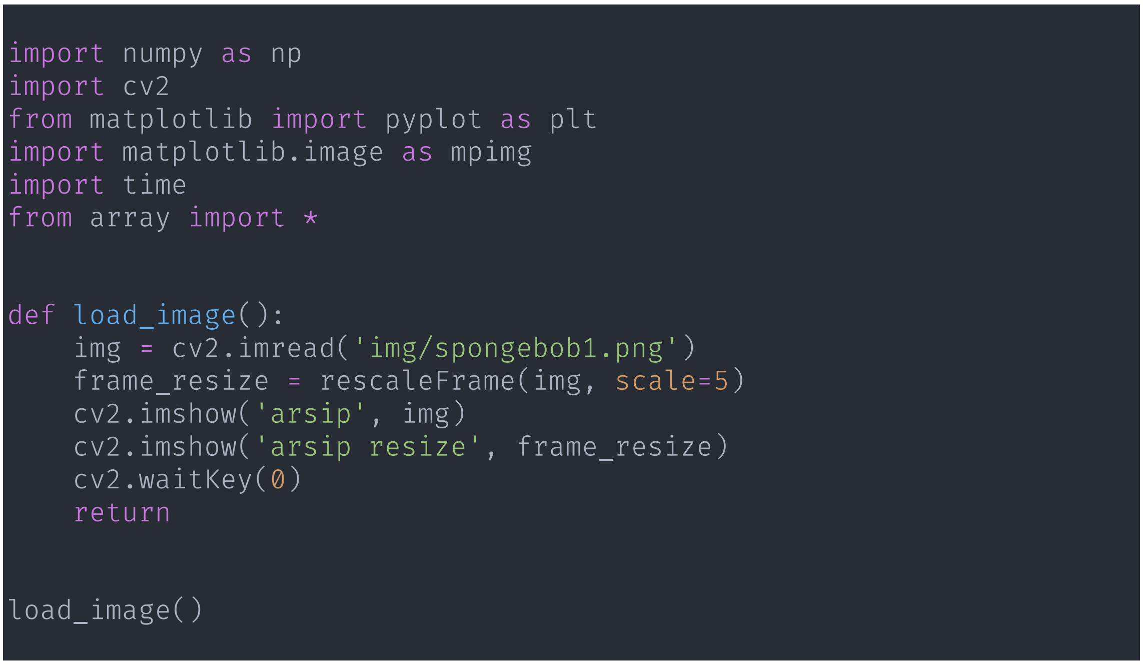
3. Install opencv menggunakan pip



```
C:\WINDOWS\system32>python
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> print(cv2.__version__)
4.5.5
>>>
```

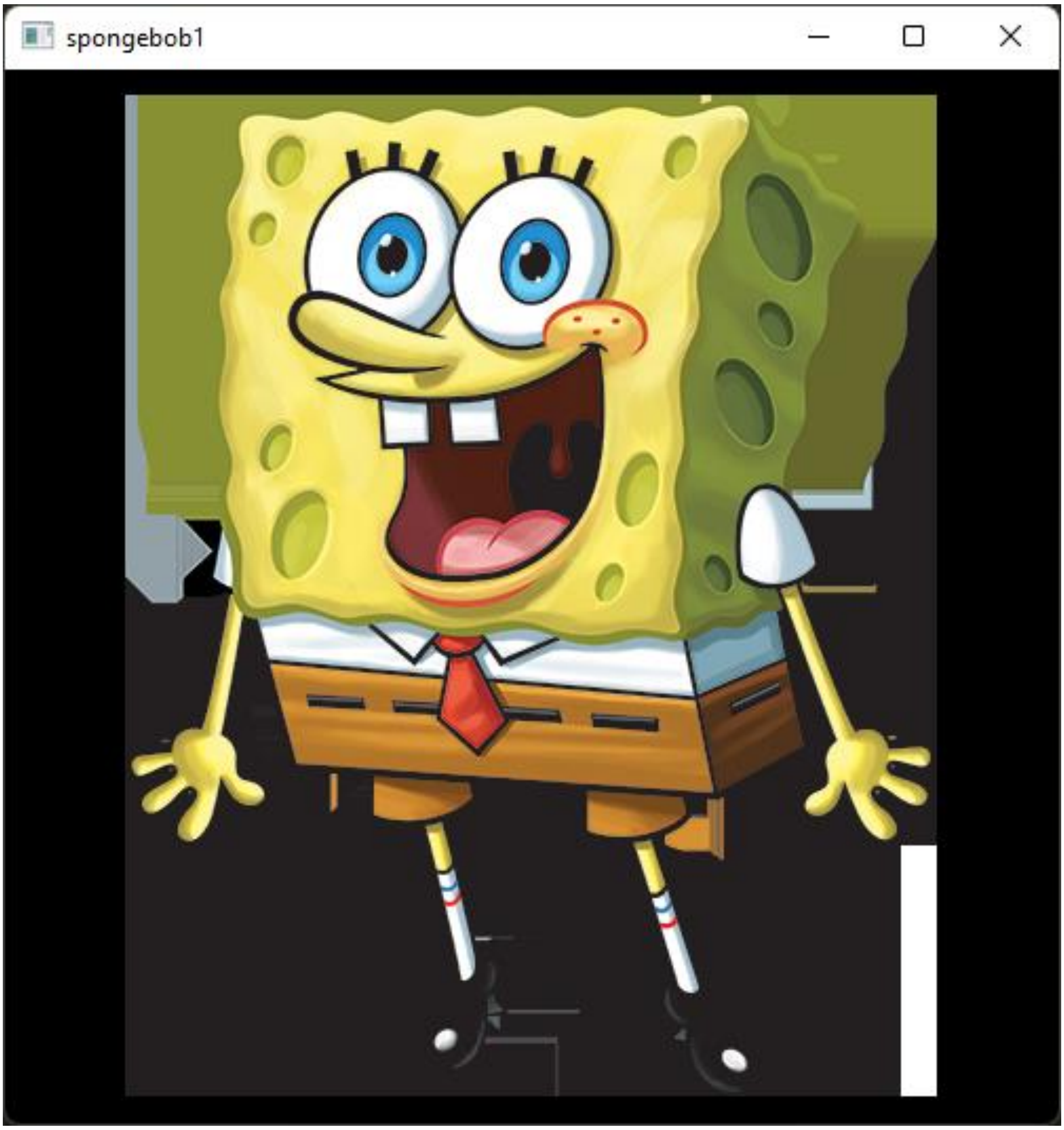
4. Menampilkan gambar

Sorce Code



```python
import numpy as np
import cv2
from matplotlib import pyplot as plt
import matplotlib.image as mpimg
import time
from array import *


def load_image():
    img = cv2.imread('img/spongebob1.png')
    frame_resize = rescaleFrame(img, scale=5)
    cv2.imshow('arsip', img)
    cv2.imshow('arsip resize', frame_resize)
    cv2.waitKey(0)
    return


load_image()
```

Output Source Code



Pengolahan Citra

Pekan 02

Dosen Pengampu

Hero Yudo Martono ST, MT

Disusun Oleh :

Nama	:	M. Faza Nur Husain
Nrp	:	3121550004

D3 PJJ AK TEKNIK INFORMATIKA

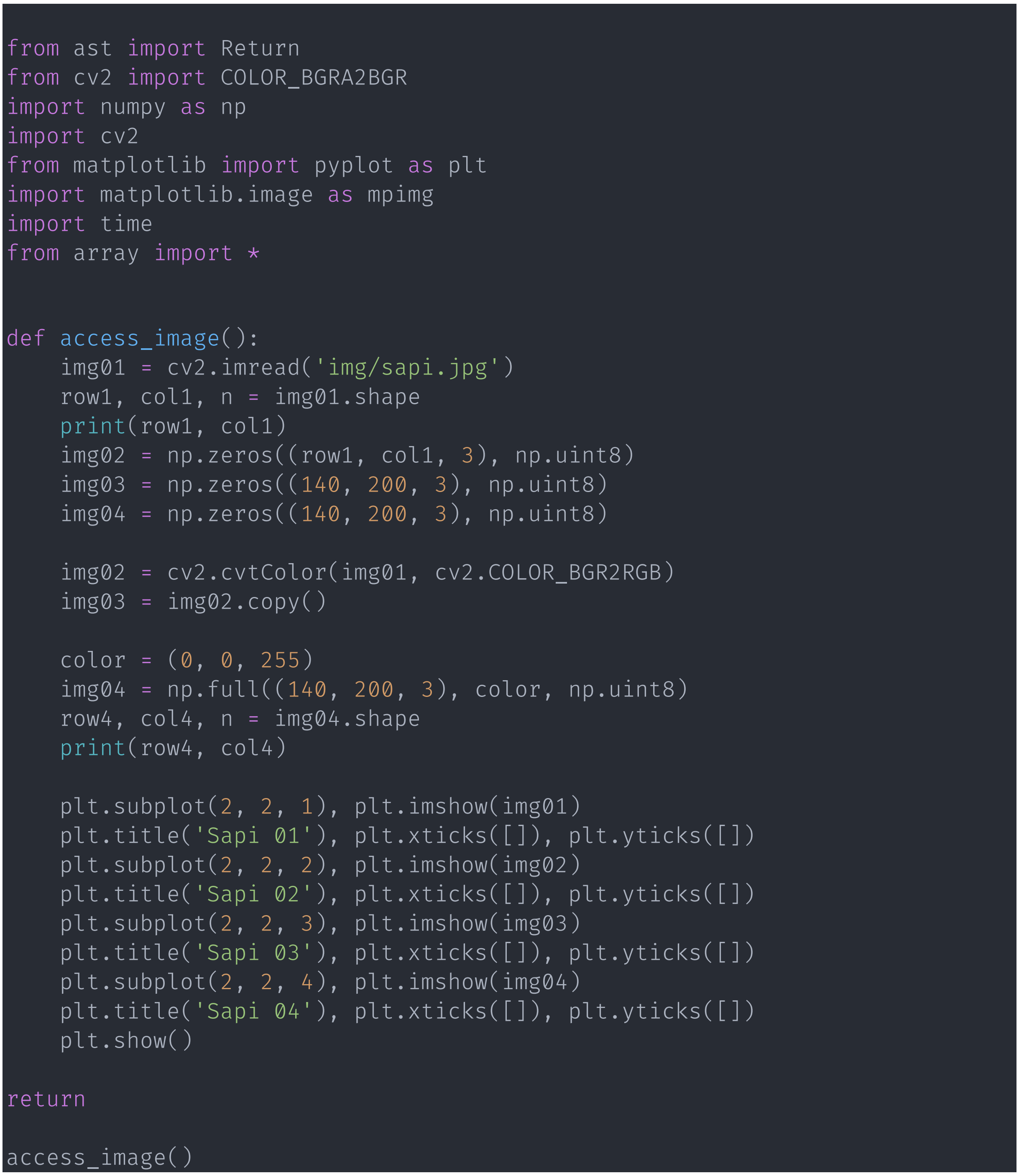
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

TAHUN AKADEMIK 2021/2022

Tugas

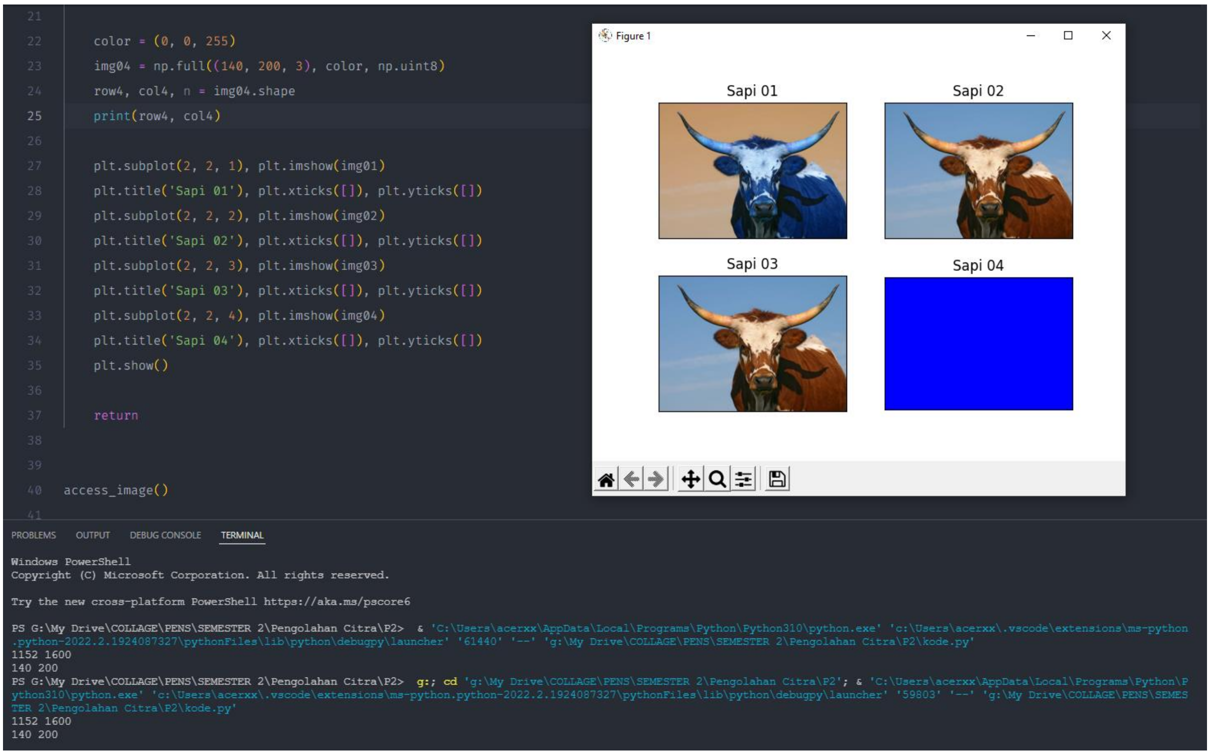
Membuat gambar / mendapatkan nilai RGB per pixle, membuat gambar sederhana : garis, kotak, segitiga, lingkaran, memberi warna pada gambar. Konversi RGB gray biner dan flip gambar horizontal dan vertical, menggunakan fingsi dari library opencv

Source Code



```
from ast import Return
from cv2 import COLOR_BGRA2BGR
import numpy as np
import cv2
from matplotlib import pyplot as plt
import matplotlib.image as mpimg
import time
from array import *


def access_image():
    img01 = cv2.imread('img/sapi.jpg')
    row1, col1, n = img01.shape
    print(row1, col1)
    img02 = np.zeros((row1, col1, 3), np.uint8)
    img03 = np.zeros((140, 200, 3), np.uint8)
    img04 = np.zeros((140, 200, 3), np.uint8)

    img02 = cv2.cvtColor(img01, cv2.COLOR_BGR2RGB)
    img03 = img02.copy()

    color = (0, 0, 255)
    img04 = np.full((140, 200, 3), color, np.uint8)
    row4, col4, n = img04.shape
    print(row4, col4)

    plt.subplot(2, 2, 1), plt.imshow(img01)
    plt.title('Sapi 01'), plt.xticks([]), plt.yticks([])
    plt.subplot(2, 2, 2), plt.imshow(img02)
    plt.title('Sapi 02'), plt.xticks([]), plt.yticks([])
    plt.subplot(2, 2, 3), plt.imshow(img03)
    plt.title('Sapi 03'), plt.xticks([]), plt.yticks([])
    plt.subplot(2, 2, 4), plt.imshow(img04)
    plt.title('Sapi 04'), plt.xticks([]), plt.yticks([])
    plt.show()

return

access_image()
```

Ouput Source Code:



```
    color = (0, 0, 255)
    img04 = np.full((140, 200, 3), color, np.uint8)
    row4, col4, n = img04.shape
    print(row4, col4)

    plt.subplot(2, 2, 1), plt.imshow(img01)
    plt.title('Sapi 01'), plt.xticks([]), plt.yticks([])
    plt.subplot(2, 2, 2), plt.imshow(img02)
    plt.title('Sapi 02'), plt.xticks([]), plt.yticks([])
    plt.subplot(2, 2, 3), plt.imshow(img03)
    plt.title('Sapi 03'), plt.xticks([]), plt.yticks([])
    plt.subplot(2, 2, 4), plt.imshow(img04)
    plt.title('Sapi 04'), plt.xticks([]), plt.yticks([])
    plt.show()

    return


access_image()
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS G:\My Drive\COLLAGE\PENS\SEMESTER 2\Pengolahan Citra\P2>  & 'C:\Users\acerxx\AppData\Local\Programs\Python\Python310\python.exe' 'c:\Users\acerxx\.vscode\extensions\ms-python
.python-2022.2.1924087327\pythonFiles\lib\python\debugpy\launcher' '61440' '--' 'g:\My Drive\COLLAGE\PENS\SEMESTER 2\Pengolahan Citra\P2\kode.py'
1152 1600
140 200
PS G:\My Drive\COLLAGE\PENS\SEMESTER 2\Pengolahan Citra\P2>  g:; cd 'g:\My Drive\COLLAGE\PENS\SEMESTER 2\Pengolahan Citra\P2'; & 'C:\Users\acerxx\AppData\Local\Programs\Python\P
ython310\python.exe' 'c:\Users\acerxx\.vscode\extensions\ms-python.python-2022.2.1924087327\pythonFiles\lib\python\debugpy\launcher' '59803' '--' 'g:\My Drive\COLLAGE\PENS\SEMES
TER 2\Pengolahan Citra\P2\kode.py'
1152 1600
140 200
```

Pengolahan Citra

Pekan 3. Enhancement : brigthness kuantisasi

Dosen Pengampu

Hero Yudo Martono ST, MT

Disusun Oleh :

Nama : M. Faza Nur Husain

Nrp : 3121550004

D3 PJJ AK TEKNIK INFORMATIKA

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

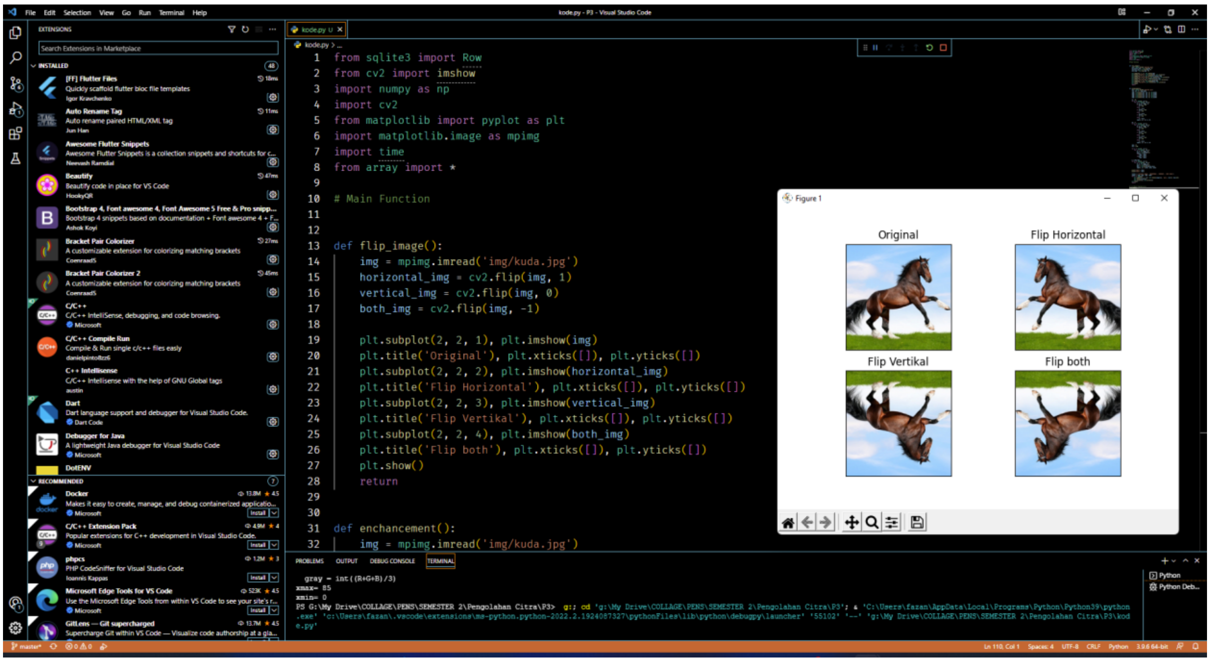
TAHUN AKADEMIK 2021/2022

Source Code :

```
from sqlite3 import Row
from cv2 import imshow
import numpy as np
import cv2
from matplotlib import pyplot as plt
import matplotlib.image as mpimg
import time
from array import *

# Main Function


def flip_image():
    img = mpimg.imread('img/kuda.jpg')
    horizontal_img = cv2.flip(img, 1)
    vertical_img = cv2.flip(img, 0)
    both_img = cv2.flip(img, -1)

    plt.subplot(2, 2, 1), plt.imshow(img)
    plt.title('Original'), plt.xticks([]), plt.yticks([])
    plt.subplot(2, 2, 2), plt.imshow(horizontal_img)
    plt.title('Flip Horizontal'), plt.xticks([]), plt.yticks([])
    plt.subplot(2, 2, 3), plt.imshow(vertical_img)
    plt.title('Flip Vertikal'), plt.xticks([]), plt.yticks([])
    plt.subplot(2, 2, 4), plt.imshow(both_img)
    plt.title('Flip both'), plt.xticks([]), plt.yticks([])
    plt.show()
    return


def enchancement():
    img = mpimg.imread('img/kuda.jpg')
    row, col, n = img.shape
    img1 = np.zeros((row, col, 3), np.uint8)
    img2 = np.zeros((row, col, 3), np.uint8)
    img3 = np.zeros((row, col, 3), np.uint8)
    img4 = np.zeros((row, col, 3), np.uint8)
    img5 = np.zeros((row, col, 3), np.uint8)

    th = 50
    for y in range(0, col-1):
        for x in range(0, row-1):
            R, G, B = img[x, y]
            if (R+th) > 255:
                R = 255
            else:
                R = R+th
            if (G+th) > 255:
                G = 255
            else:
                G = G+th
            if (R+th) > 255:
                R = 255
            else:
                R = R+th
            img1[x, y] = [R, G, B]

    th = 4
    for y in range(0, col-1):
        for x in range(0, row-1):
            R, G, B = img[x, y]
            if (R*th) > 255:
                R = 255
            else:
                R = R*th
            if (G*th) > 255:
                G = 255
            else:
                G = G*th
            if (R*th) > 255:
                R = 255
            else:
                R = R*th
            img2[x, y] = [R, G, B]

    xmax = 0
    xmin = 300

    for y in range(0, col-1):
        for x in range(0, row-1):
            R, G, B = img[x, y]
            gray = int((R+G+B)/3)
            if(gray > xmax):
                xmax = gray
            if(gray < xmin):
                xmin = gray

    d = xmax-xmin
    for y in range(0, col-1):
        for x in range(0, row-1):
            R, G, B = img[x, y]
            gray = int((R+G+B)/3)
            gray = int((255/d)*gray-xmin)
            img3[x, y] = [gray, gray, gray]

    print("xmax=", xmax)
    print("xmin=", xmin)

    titles = ['Original Image', 'BRIGHTNESS', 'CONTRAST', 'AUTO SCALE']
    images = [img, img1, img2, img3]
    for i in range(4):
        plt.subplot(2, 2, i+1), plt.imshow(images[i], 'gray', vmin=0,
vmax=255)
        plt.title(titles[i])
        plt.xticks([]), plt.yticks([])
    plt.show()
    return


# ======Main Program=======
# flip_image()
# enchancement()
```

Output Code :

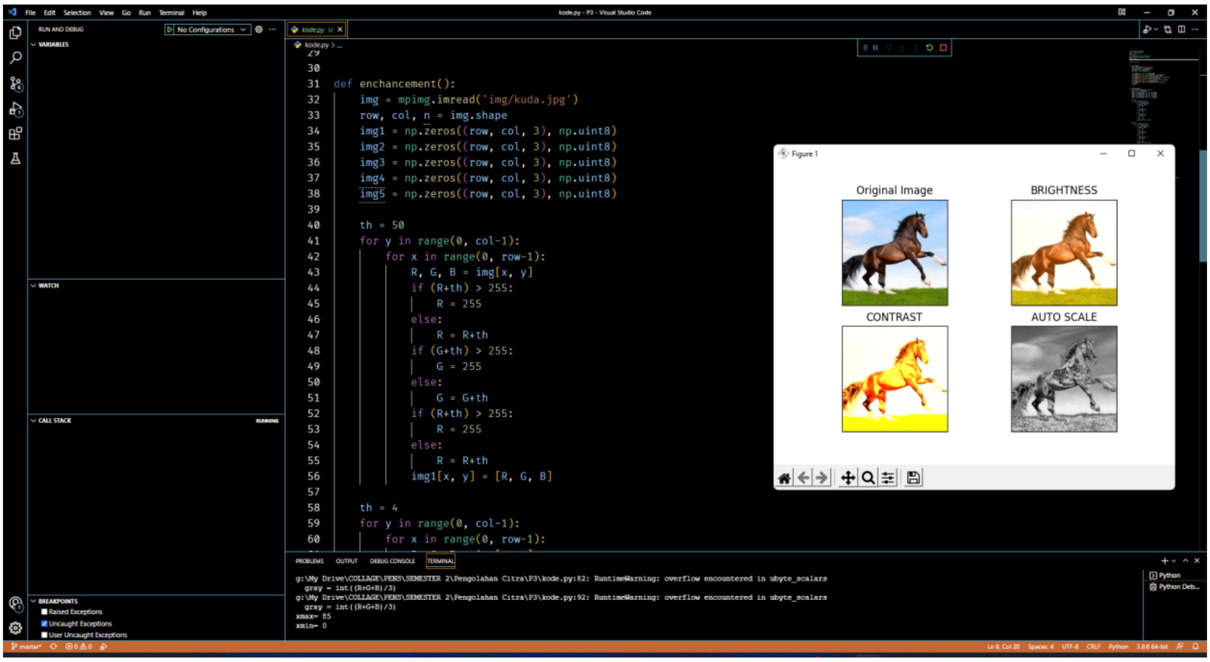
flip_image()



```
from sqlite3 import Row
from cv2 import imshow
import numpy as np
import cv2
from matplotlib import pyplot as plt
import matplotlib.image as mpimg
import time
from array import *

# Main Function


def flip_image():
    img = mpimg.imread('img/kuda.jpg')
    horizontal_img = cv2.flip(img, 1)
    vertical_img = cv2.flip(img, 0)
    both_img = cv2.flip(img, -1)

    plt.subplot(2, 2, 1), plt.imshow(img)
    plt.title('Original'), plt.xticks([]), plt.yticks([])
    plt.subplot(2, 2, 2), plt.imshow(horizontal_img)
    plt.title('Flip Horizontal'), plt.xticks([]), plt.yticks([])
    plt.subplot(2, 2, 3), plt.imshow(vertical_img)
    plt.title('Flip Vertikal'), plt.xticks([]), plt.yticks([])
    plt.subplot(2, 2, 4), plt.imshow(both_img)
    plt.title('Flip both'), plt.xticks([]), plt.yticks([])
    plt.show()
    return


def enchancement():
    img = mpimg.imread('img/kuda.jpg')
```

```
  gray = int((R+G+B)/3)
xmax= 85
xmin= 0
PS G:\My Drive\COLLAGE\PENS\SEMESTER 2\Pengolahan Citra\P3>  g:; cd 'g:\My Drive\COLLAGE\PENS\SEMESTER 2\Pengolahan Citra\P3'; & 'C:\Users\fazan\AppData\Local\Programs\Python\Python39\python.exe' 'c:\Users\fazan\.vscode\extensions\ms-python.python-2022.2.1924087327\pythonFiles\lib\python\debugpy\launcher' '55102' '--' 'g:\My Drive\COLLAGE\PENS\SEMESTER 2\Pengolahan Citra\P3\kode.py'
```

enchancement()



```
def enchancement():
    img = mpimg.imread('img/kuda.jpg')
    row, col, n = img.shape
    img1 = np.zeros((row, col, 3), np.uint8)
    img2 = np.zeros((row, col, 3), np.uint8)
    img3 = np.zeros((row, col, 3), np.uint8)
    img4 = np.zeros((row, col, 3), np.uint8)
    img5 = np.zeros((row, col, 3), np.uint8)

    th = 50
    for y in range(0, col-1):
        for x in range(0, row-1):
            R, G, B = img[x, y]
            if (R+th) > 255:
                R = 255
            else:
                R = R+th
            if (G+th) > 255:
                G = 255
            else:
                G = G+th
            if (R+th) > 255:
                R = 255
            else:
                R = R+th
            img1[x, y] = [R, G, B]

    th = 4
    for y in range(0, col-1):
        for x in range(0, row-1):
```

```
g:\My Drive\COLLAGE\PENS\SEMESTER 2\Pengolahan Citra\P3\kode.py:82: RuntimeWarning: overflow encountered in ubyte_scalars
  gray = int((R+G+B)/3)
g:\My Drive\COLLAGE\PENS\SEMESTER 2\Pengolahan Citra\P3\kode.py:92: RuntimeWarning: overflow encountered in ubyte_scalars
  gray = int((R+G+B)/3)
xmax= 85
xmin= 0
```

Pengolahan Citra

Pekan 4. Filtering LPH dan HPF in domain spatial

Dosen Pengampu

Hero Yudo Martono ST, MT

Disusun Oleh :

Nama	:	M. Faza Nur Husain
Nrp	:	3121550004

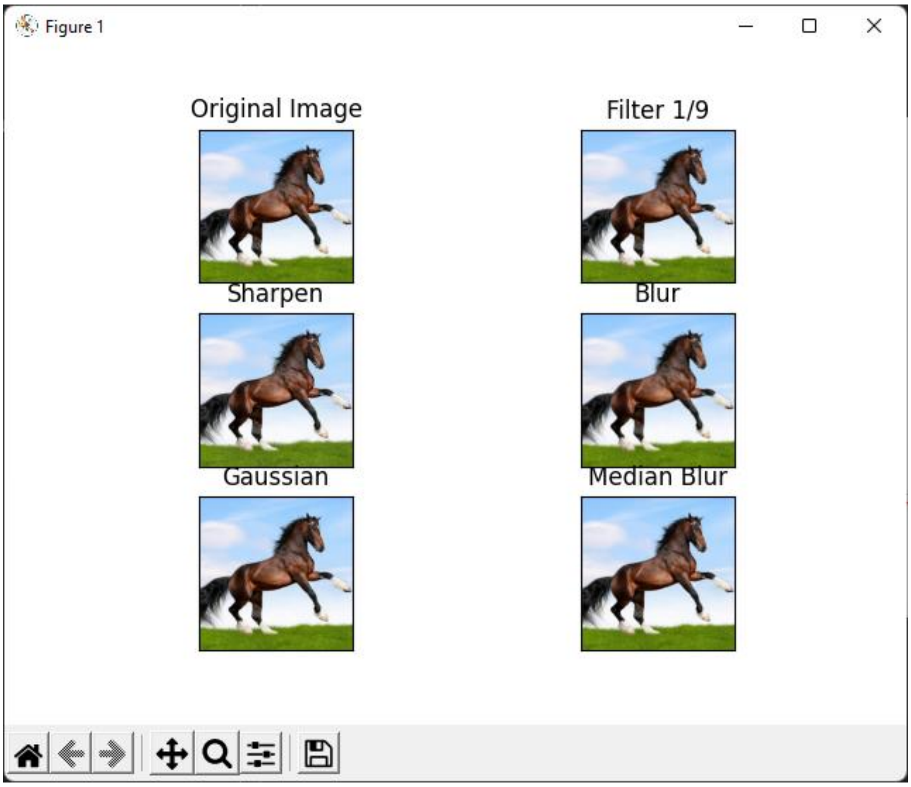
D3 PJJ AK TEKNIK INFORMATIKA

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

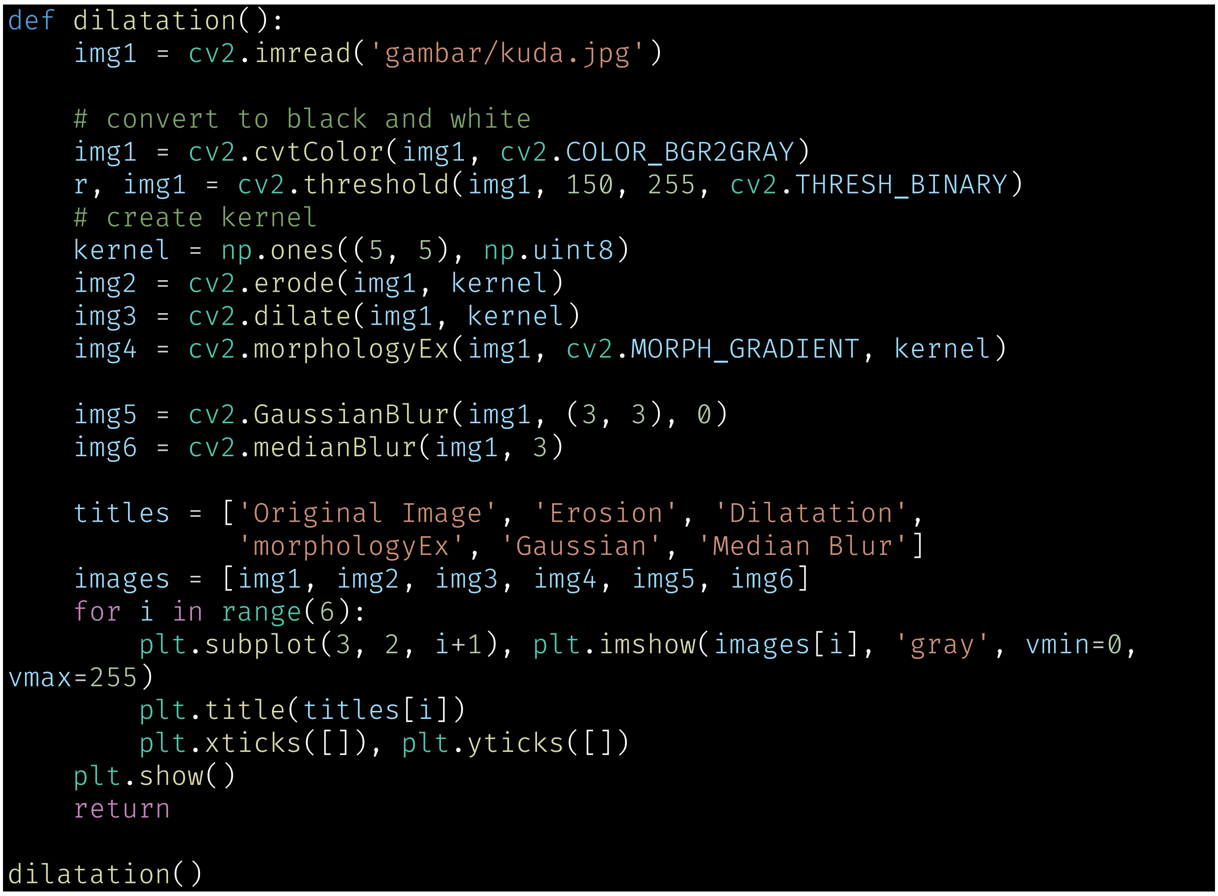
TAHUN AKADEMIK 2021/2022

Source Code :

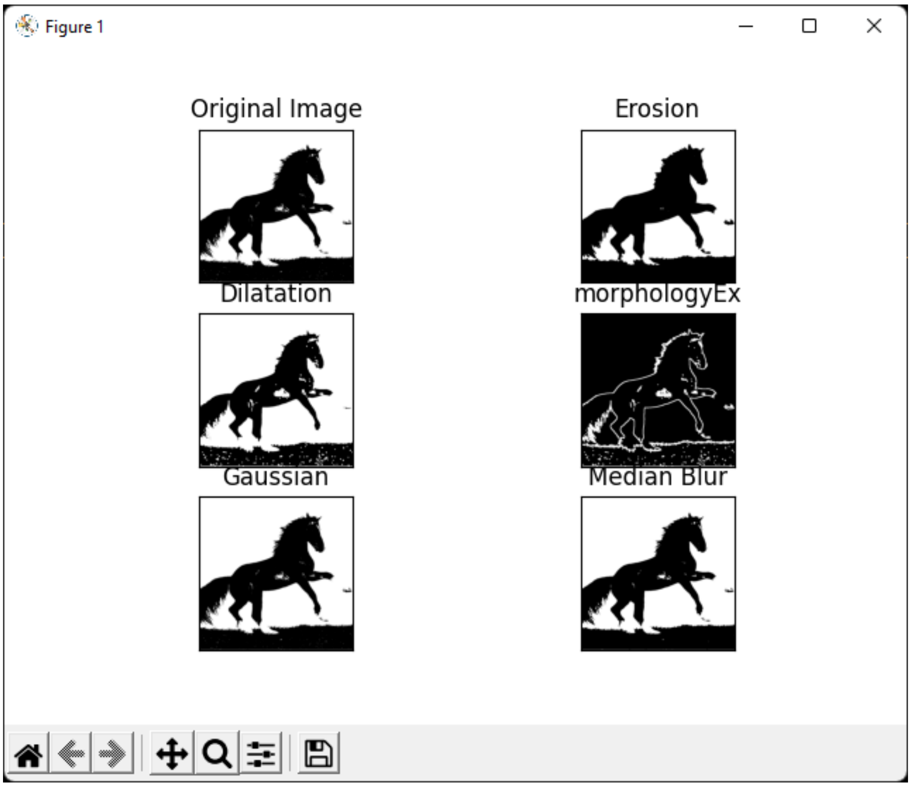
```
def convolution2D():
    img1 = cv2.imread('gambar/kuda.jpg')
    img1 = cv2.cvtColor(img1, cv2.COLOR_BGR2RGB)
    kernel = np.ones((3, 3), np.float32) / 9
    #print (kernel)
    img2 = cv2.filter2D(img1, -1, kernel)

    kernel = np.array([[0,  -1, 0],
                       [-1, 5, -1],
                       [0, -1, 0]])
    img3 = cv2.filter2D(img1, -1, kernel)

    img4 = cv2.blur(img1, (5, 5))
    img5 = cv2.GaussianBlur(img1, (3, 3), 0)
    img6 = cv2.medianBlur(img1, 3)

    titles = ['Original Image', 'Filter 1/9',
              'Sharpen', 'Blur', 'Gaussian', 'Median Blur']
    images = [img1, img2, img3, img4, img5, img6]
    for i in range(6):
        plt.subplot(3, 2, i+1), plt.imshow(images[i], 'gray', vmin=0,
vmax=255)
        plt.title(titles[i])
        plt.xticks([]), plt.yticks([])
    plt.show()
    return
convolution2D()
```



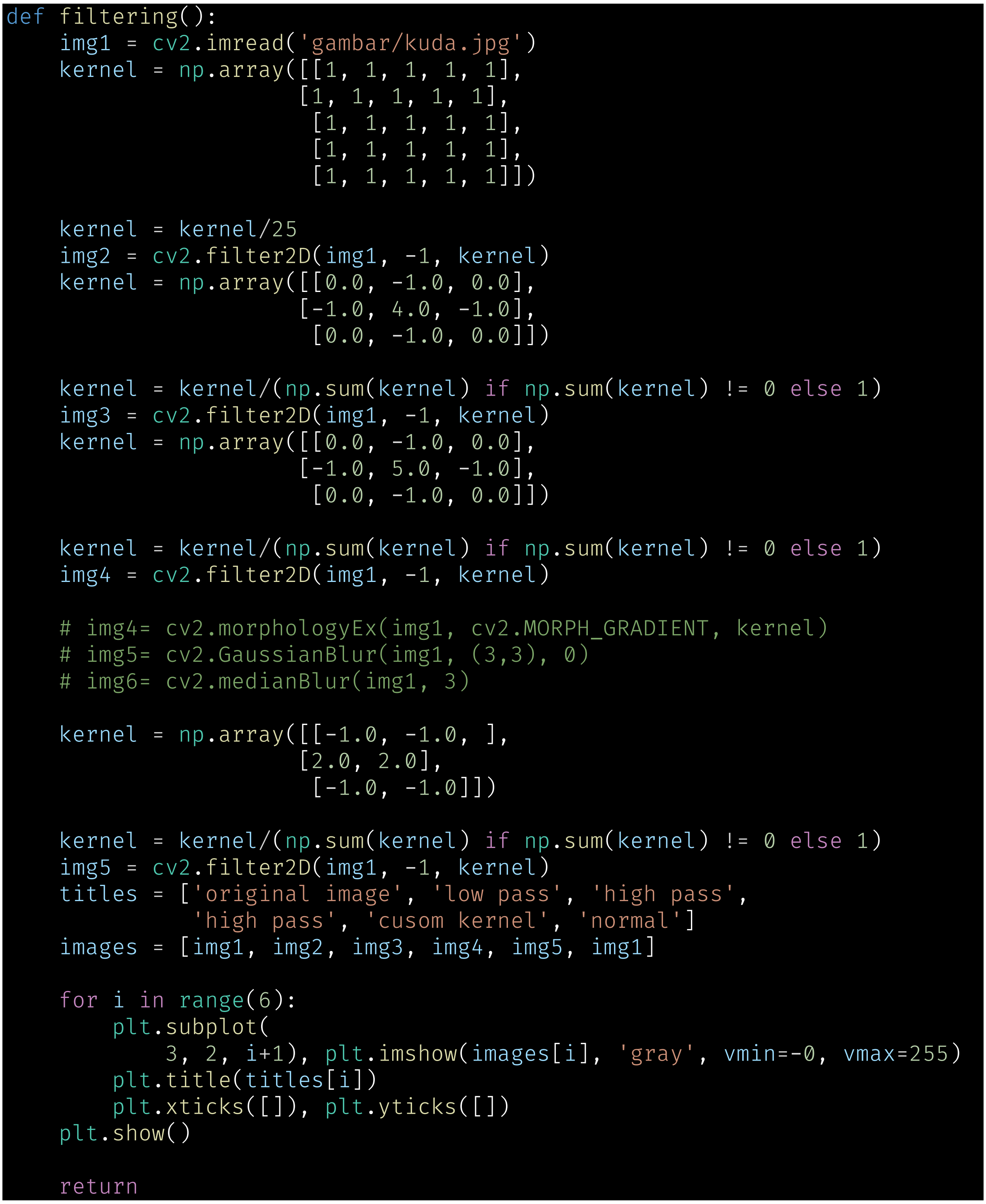
Source Code :



```
def dilatation():
    img1 = cv2.imread('gambar/kuda.jpg')

    # convert to black and white
    img1 = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
    r, img1 = cv2.threshold(img1, 150, 255, cv2.THRESH_BINARY)
    # create kernel
    kernel = np.ones((5, 5), np.uint8)
    img2 = cv2.erode(img1, kernel)
    img3 = cv2.dilate(img1, kernel)
    img4 = cv2.morphologyEx(img1, cv2.MORPH_GRADIENT, kernel)

    img5 = cv2.GaussianBlur(img1, (3, 3), 0)
    img6 = cv2.medianBlur(img1, 3)

    titles = ['Original Image', 'Erosion', 'Dilatation',
              'morphologyEx', 'Gaussian', 'Median Blur']
    images = [img1, img2, img3, img4, img5, img6]
    for i in range(6):
        plt.subplot(3, 2, i+1), plt.imshow(images[i], 'gray', vmin=0,
vmax=255)
        plt.title(titles[i])
        plt.xticks([]), plt.yticks([])
    plt.show()
    return

dilatation()
```



Source Code :



```
def filtering():
    img1 = cv2.imread('gambar/kuda.jpg')
    kernel = np.array([[1, 1, 1, 1, 1],
                       [1, 1, 1, 1, 1],
                        [1, 1, 1, 1, 1],
                        [1, 1, 1, 1, 1],
                        [1, 1, 1, 1, 1]])

    kernel = kernel/25
    img2 = cv2.filter2D(img1, -1, kernel)
    kernel = np.array([[0.0, -1.0, 0.0],
                       [-1.0, 4.0, -1.0],
                        [0.0, -1.0, 0.0]])

    kernel = kernel/(np.sum(kernel) if np.sum(kernel) != 0 else 1)
    img3 = cv2.filter2D(img1, -1, kernel)
    kernel = np.array([[0.0, -1.0, 0.0],
                       [-1.0, 5.0, -1.0],
                        [0.0, -1.0, 0.0]])

    kernel = kernel/(np.sum(kernel) if np.sum(kernel) != 0 else 1)
    img4 = cv2.filter2D(img1, -1, kernel)

    # img4= cv2.morphologyEx(img1, cv2.MORPH_GRADIENT, kernel)
    # img5= cv2.GaussianBlur(img1, (3,3), 0)
    # img6= cv2.medianBlur(img1, 3)

    kernel = np.array([[-1.0, -1.0, ],
                       [2.0, 2.0],
                        [-1.0, -1.0]])

    kernel = kernel/(np.sum(kernel) if np.sum(kernel) != 0 else 1)
    img5 = cv2.filter2D(img1, -1, kernel)
    titles = ['original image', 'low pass', 'high pass',
              'high pass', 'cusom kernel', 'normal']
    images = [img1, img2, img3, img4, img5, img1]

    for i in range(6):
        plt.subplot(
            3, 2, i+1), plt.imshow(images[i], 'gray', vmin=-0, vmax=255)
        plt.title(titles[i])
        plt.xticks([]), plt.yticks([])
    plt.show()

    return
```

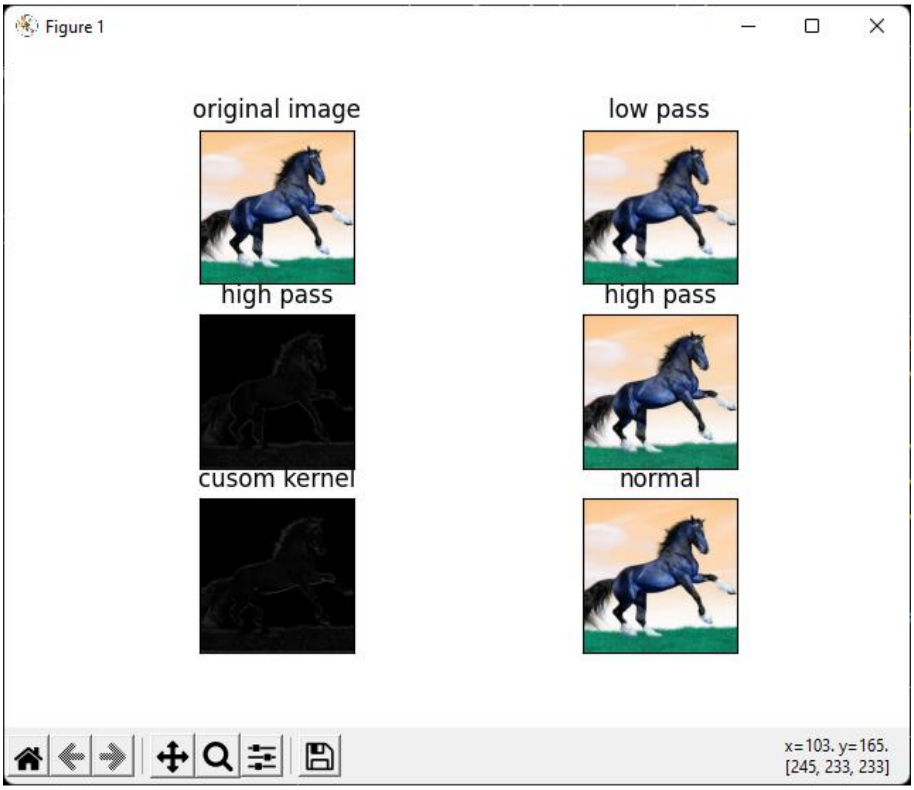


Figure 1
original image
low pass
high pass
high pass
cusom kernel
normal
x=103. y=165.
[245, 233, 233]

Pengolahan Citra

5. Filtering LPH dan HPF in domain frequency

Dosen Pengampu

Hero Yudo Martono ST, MT

Disusun Oleh :

Nama : M. Faza Nur Husain

Nrp : 3121550004

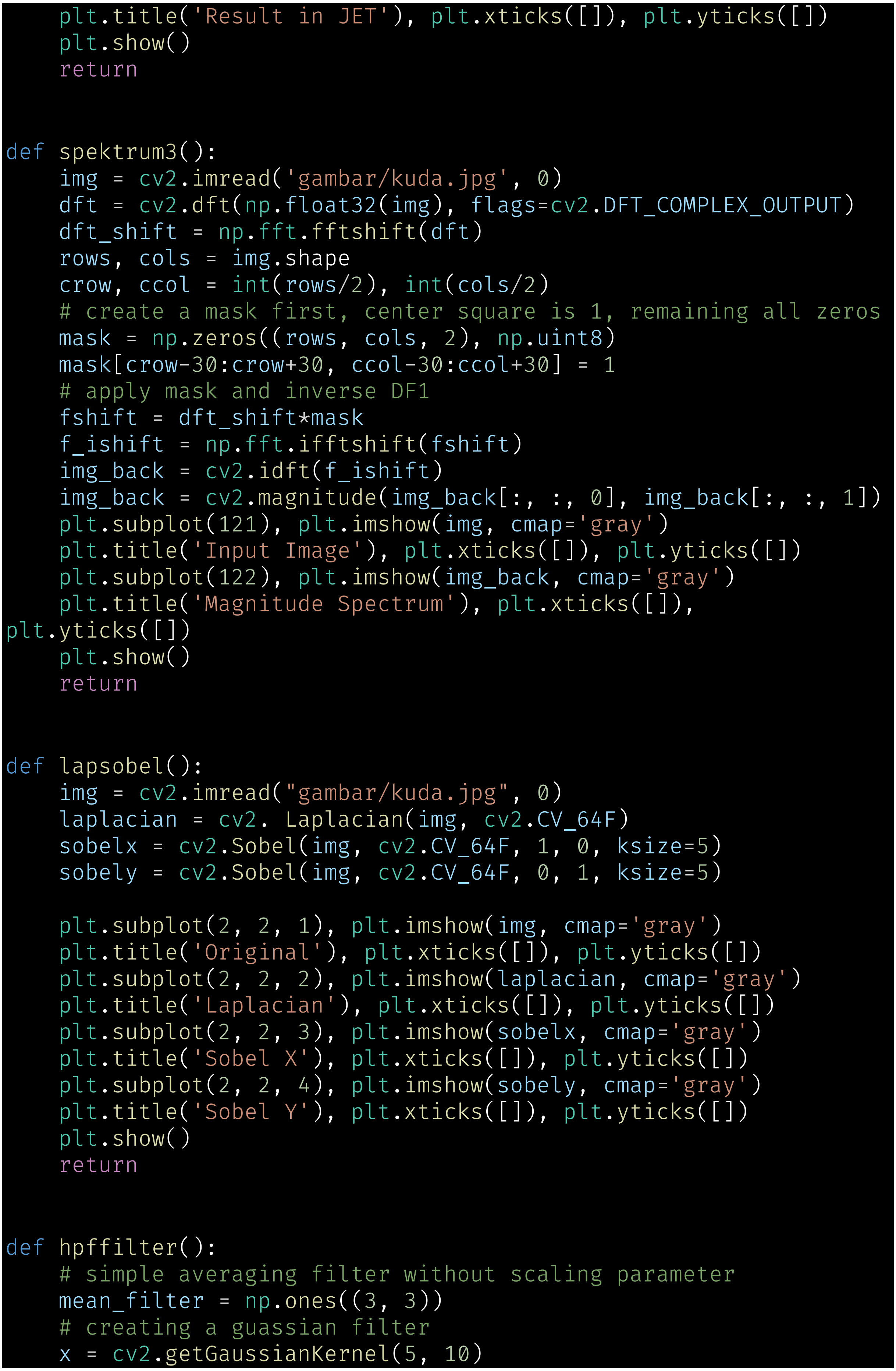
D3 PJJ AK TEKNIK INFORMATIKA

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

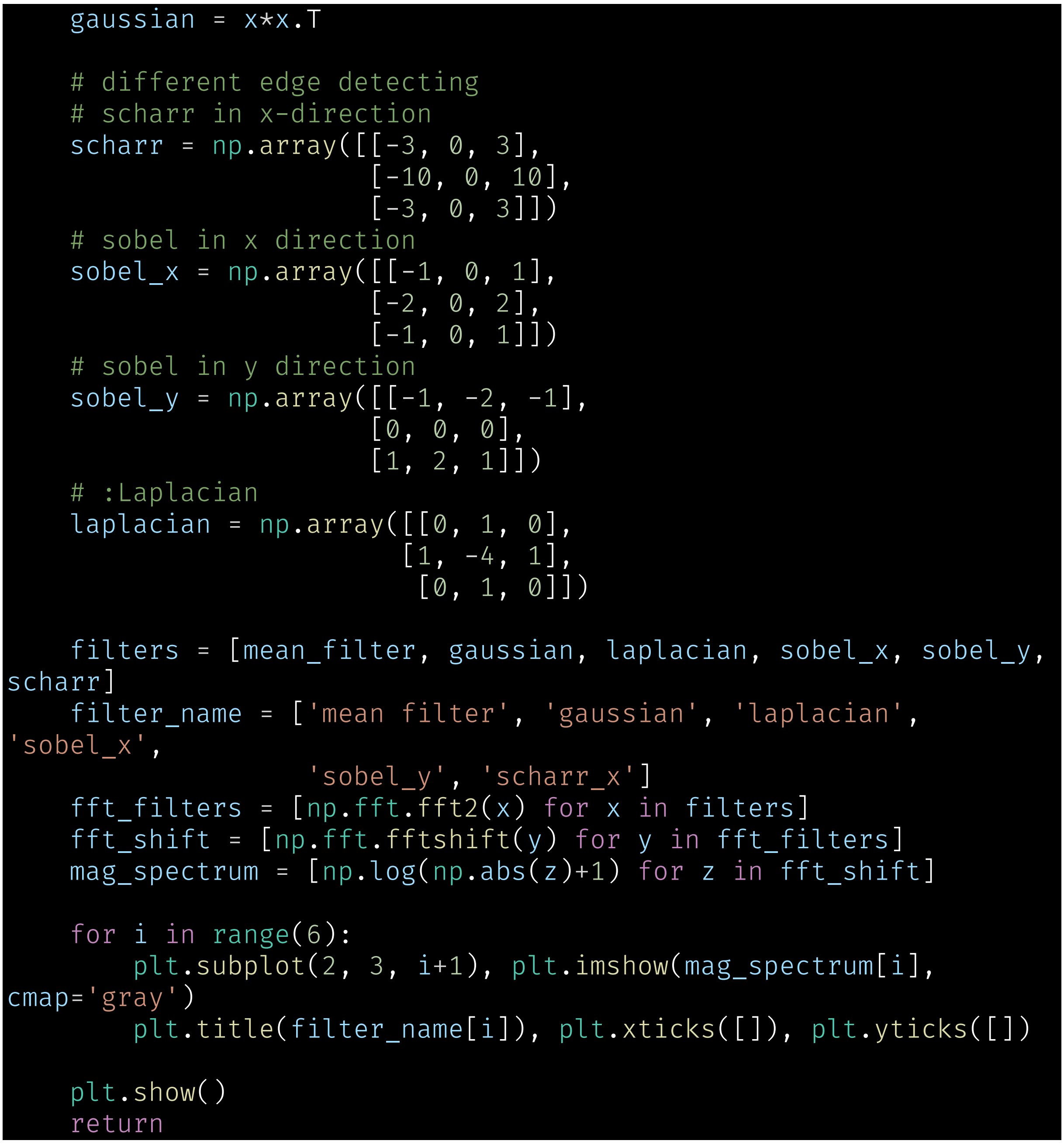
TAHUN AKADEMIK 2021/2022

Source Code :

```
def spektrum():
    img = cv2.imread('gambar/kuda.jpg', 0)
    img_float32 = np.float32(img)
    dft = cv2.dft(img_float32, flags=cv2.DFT_COMPLEX_OUTPUT)
    dft_shift = np.fft.fftshift(dft)
    magnitude_spectrum = 20 * \
        np.log(cv2.magnitude(dft_shift[:, :, 0], dft_shift[:, :,
1]))
    plt.subplot(121), plt.imshow(img, cmap='gray')
    plt.title('Input Image'), plt.xticks([]), plt.yticks([])
    plt.subplot(122), plt.imshow(magnitude_spectrum, cmap='gray')
    plt.title('Magnitude Spectrum'), plt.xticks([]),
plt.yticks([])
    plt.show()
    return


def spektrum2():
    img = cv2.imread('gambar/kuda.jpg', 0)
    f = np.fft.fft2(img)
    fshift = np.fft.fftshift(f)
    magnitude_spectrum = 20*np.log(np.abs(fshift))
    plt.subplot(121), plt.imshow(img, cmap='gray')
    plt.title('Input Image'), plt.xticks([]), plt.yticks([])
    plt.subplot(122), plt.imshow(magnitude_spectrum, cmap='gray')
    plt.title('Magnitude Spectrum'), plt.xticks([]),
plt.yticks([])
    plt.show()
    return


def afterhpfjet():
    img = cv2.imread('gambar/kuda.jpg', 0)
    f = np.fft.fft2(img)
    fshift = np.fft.fftshift(f)
    magnitude_spectrum = 20*np.log(np.abs(fshift))
    rows, cols = img.shape
    crow, ccol = int(rows/2), int(cols/2)
    print(crow, ccol)
    fshift[crow-30:crow+30, ccol-30:ccol+30] = 0
    f_ishift = np.fft.ifftshift(fshift)
    img_back = np.fft.ifft2(f_ishift)
    img_back = np.abs(img_back)

    plt.subplot(131), plt.imshow(img, cmap='gray')
    plt.title('Input Image'), plt.xticks([]), plt.yticks([])
    plt.subplot(132), plt.imshow(img_back, cmap='gray')
    plt.title('Image after HPF'), plt.xticks([]), plt.yticks([])
    plt.subplot(133), plt.imshow(img_back)
```



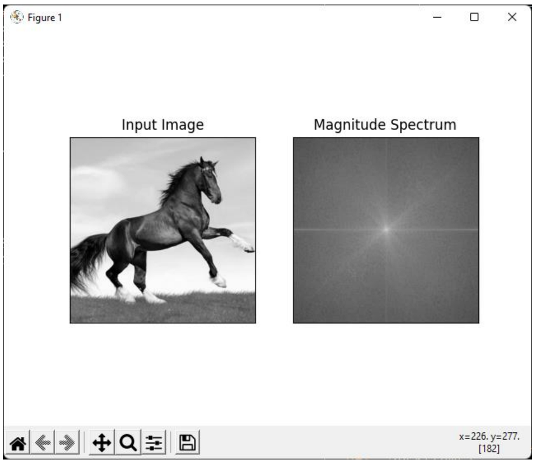
```
    plt.title('Result in JET'), plt.xticks([]), plt.yticks([])
    plt.show()
    return


def spektrum3():
    img = cv2.imread('gambar/kuda.jpg', 0)
    dft = cv2.dft(np.float32(img), flags=cv2.DFT_COMPLEX_OUTPUT)
    dft_shift = np.fft.fftshift(dft)
    rows, cols = img.shape
    crow, ccol = int(rows/2), int(cols/2)
    # create a mask first, center square is 1, remaining all zeros
    mask = np.zeros((rows, cols, 2), np.uint8)
    mask[crow-30:crow+30, ccol-30:ccol+30] = 1
    # apply mask and inverse DF1
    fshift = dft_shift*mask
    f_ishift = np.fft.ifftshift(fshift)
    img_back = cv2.idft(f_ishift)
    img_back = cv2.magnitude(img_back[:, :, 0], img_back[:, :, 1])
    plt.subplot(121), plt.imshow(img, cmap='gray')
    plt.title('Input Image'), plt.xticks([]), plt.yticks([])
    plt.subplot(122), plt.imshow(img_back, cmap='gray')
    plt.title('Magnitude Spectrum'), plt.xticks([]),
plt.yticks([])
    plt.show()
    return


def lapsobel():
    img = cv2.imread("gambar/kuda.jpg", 0)
    laplacian = cv2. Laplacian(img, cv2.CV_64F)
    sobelx = cv2.Sobel(img, cv2.CV_64F, 1, 0, ksize=5)
    sobely = cv2.Sobel(img, cv2.CV_64F, 0, 1, ksize=5)

    plt.subplot(2, 2, 1), plt.imshow(img, cmap='gray')
    plt.title('Original'), plt.xticks([]), plt.yticks([])
    plt.subplot(2, 2, 2), plt.imshow(laplacian, cmap='gray')
    plt.title('Laplacian'), plt.xticks([]), plt.yticks([])
    plt.subplot(2, 2, 3), plt.imshow(sobelx, cmap='gray')
    plt.title('Sobel X'), plt.xticks([]), plt.yticks([])
    plt.subplot(2, 2, 4), plt.imshow(sobely, cmap='gray')
    plt.title('Sobel Y'), plt.xticks([]), plt.yticks([])
    plt.show()
    return


def hpffilter():
    # simple averaging filter without scaling parameter
    mean_filter = np.ones((3, 3))
    # creating a guassian filter
    x = cv2.getGaussianKernel(5, 10)
```



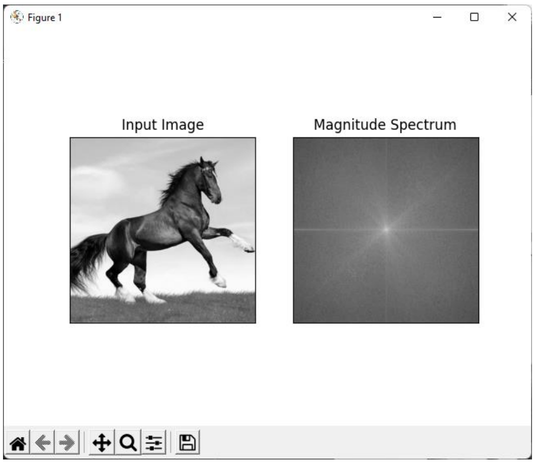
```
    gaussian = x*x.T

    # different edge detecting
    # scharr in x-direction
    scharr = np.array([[-3, 0, 3],
                       [-10, 0, 10],
                       [-3, 0, 3]])
    # sobel in x direction
    sobel_x = np.array([[-1, 0, 1],
                       [-2, 0, 2],
                       [-1, 0, 1]])
    # sobel in y direction
    sobel_y = np.array([[-1, -2, -1],
                       [0, 0, 0],
                       [1, 2, 1]])
    # :Laplacian
    laplacian = np.array([[0, 1, 0],
                         [1, -4, 1],
                          [0, 1, 0]])

    filters = [mean_filter, gaussian, laplacian, sobel_x, sobel_y,
scharr]
    filter_name = ['mean filter', 'gaussian', 'laplacian',
'sobel_x',
                   'sobel_y', 'scharr_x']
    fft_filters = [np.fft.fft2(x) for x in filters]
    fft_shift = [np.fft.fftshift(y) for y in fft_filters]
    mag_spectrum = [np.log(np.abs(z)+1) for z in fft_shift]

    for i in range(6):
        plt.subplot(2, 3, i+1), plt.imshow(mag_spectrum[i],
cmap='gray')
        plt.title(filter_name[i]), plt.xticks([]), plt.yticks([])

    plt.show()
    return
```

Output :

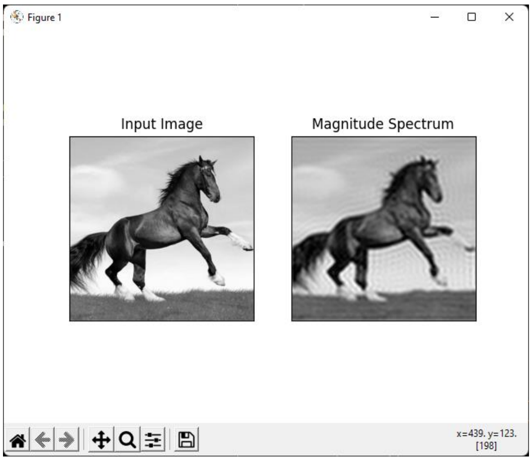
spektrum()



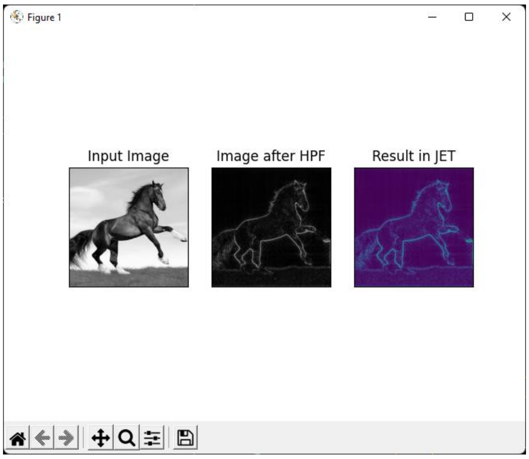
spektrum2()



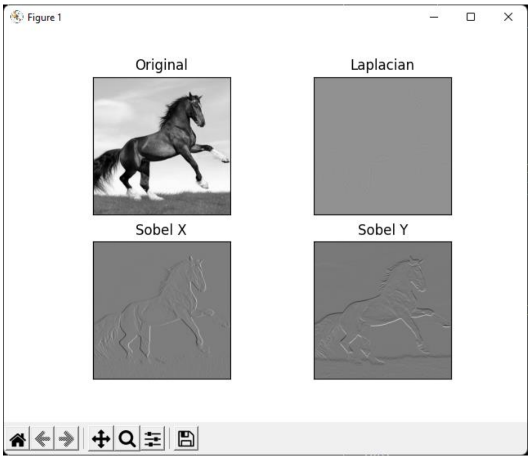
spektrum3()



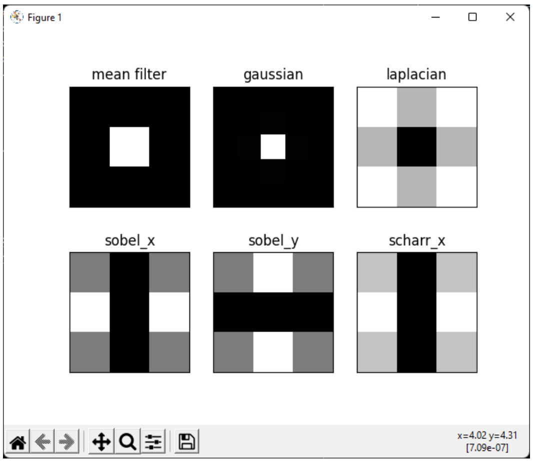
afterhpfjet()



lapsobel()



hpffilter()



https://github.com/FazaZas/pengolahan_citra.git