

**DOMAIN FREKUENSI
PENGOLAHAN CITRA DIGITAL**



Sitti Nurlaili Qofifa

F55120107

C

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
JURURSAN TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS TADULAKO**

2022

A. Tujuan

1. Mahasiswa mampu memahami apa itu Domain Frekuensi.
2. Mahasiswa mampu membuat program sederhana Perbaikan Citra Domain Frekuensi.
3. Mahasiswa mampu mengimplementasikan program Perbaikan Citra Domain Frekuensi.

B. Teori Dasar

Domain frekuensi adalah metode yang digunakan untuk menganalisis data. Ini mengacu pada analisis fungsi matematis atau sinyal berkenaan dengan frekuensi. Analisis domain frekuensi banyak digunakan di bidang-bidang seperti teknik sistem kontrol, elektronika dan statistik. Representasi domain frekuensi juga dapat mencakup informasi tentang pergeseran fasa yang harus diterapkan pada setiap sinusoida agar dapat menggabungkan kembali komponen frekuensi untuk memulihkan sinyal waktu asli. Analisis domain frekuensi sebagian besar digunakan untuk sinyal atau fungsi yang periodik dari waktu ke waktu. Ini tidak berarti bahwa analisis domain frekuensi tidak dapat digunakan dalam sinyal yang tidak periodik. Konsep yang paling penting dalam analisis domain frekuensi adalah transformasi. Transformasi digunakan untuk mengubah fungsi domain waktu menjadi fungsi domain frekuensi dan sebaliknya. Transformasi yang paling umum digunakan dalam domain frekuensi adalah Transformasi Fourier.

Transformasi Fourier adalah alat pengolah gambar penting yang digunakan untuk menguraikan gambar menjadi komponen sinus dan kosinusnya. Keluaran dari transformasi merepresentasikan citra dalam domain frekuensi, sedangkan citra masukan merupakan ekuivalen domain spasial. Pada citra domain frekuensi, setiap titik mewakili frekuensi tertentu yang terdapat pada citra domain spasial. Transformasi Fourier digunakan untuk mengubah sinyal dari bentuk apapun menjadi jumlah gelombang sinusoidal yang tak terbatas. Karena menganalisa fungsi sinusoidal lebih mudah daripada

menganalisa fungsi berbentuk umum, metode ini sangat berguna dan banyak digunakan.

Dalam Transformasi Fourier terdapat dua jenis sampel diantaranya Discrete Fourier Transform (DFT) dan Fast Fourier Transform (FFT). Discrete Fourier Transform (DFT) adalah sampel Fourier Transform, sehingga tidak mengandung semua frekuensi yang membentuk gambar, tetapi hanya satu set sampel yang cukup besar untuk sepenuhnya menggambarkan gambar domain spasial. Jumlah frekuensi sesuai dengan jumlah piksel pada citra domain spasial. Sedangkan Fast Fourier Transform (FFT) untuk menghitung DFT satu dimensi. Ini adalah peningkatan yang signifikan, khususnya untuk gambar besar. Ada berbagai bentuk FFT dan sebagian besar membatasi ukuran gambar input yang dapat ditransformasikan, seringkali ke persamaan $N = 2^n$ di mana n adalah bilangan bulat.

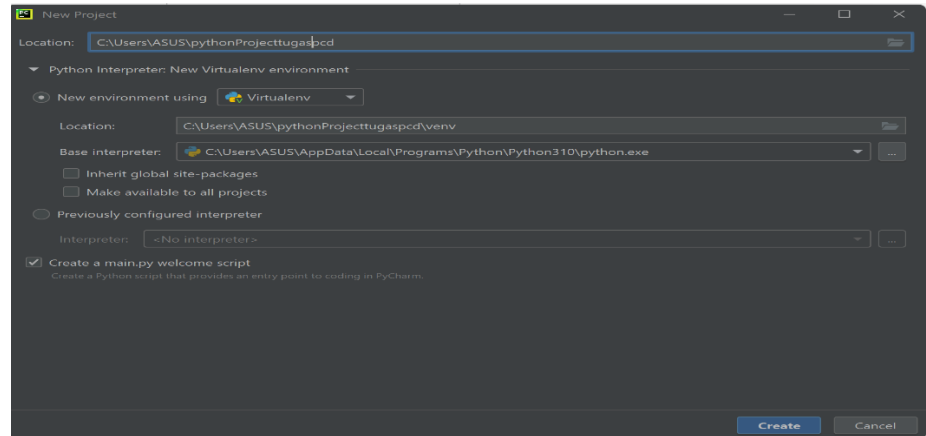
C. Praktikum

1. Sebelum memulai *coding*, pastikan sudah menginstal beberapa *library* yang diperlukan antara lain :
 - a. *Opencv-Python*.
 - b. *Numpy*.
 - c. *Matplotlib*
2. Kemudian tambahkan sebuah gambar yang akan digunakan kedalam folder *img* pada program Domain Frekuensi. Serta perhatikan Ekstensi file gambar yang ditambahkan.

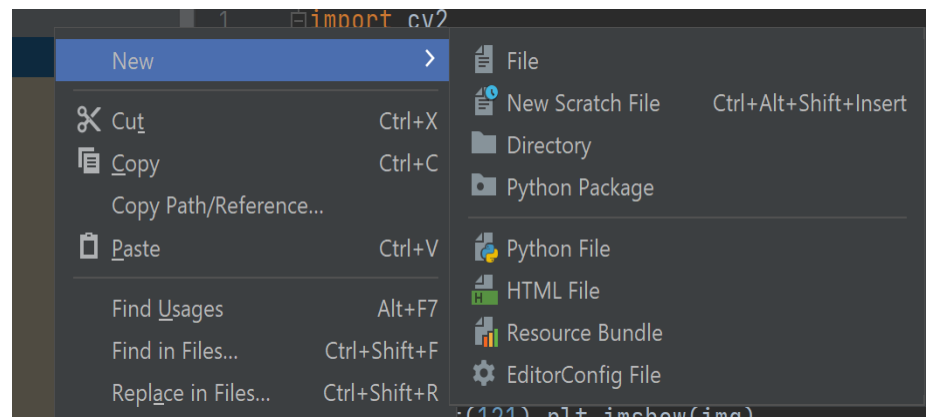


3. Selanjutnya masuk pada program Domain Frekuensi. Adapun Langkah-langkahnya sebagai berikut :

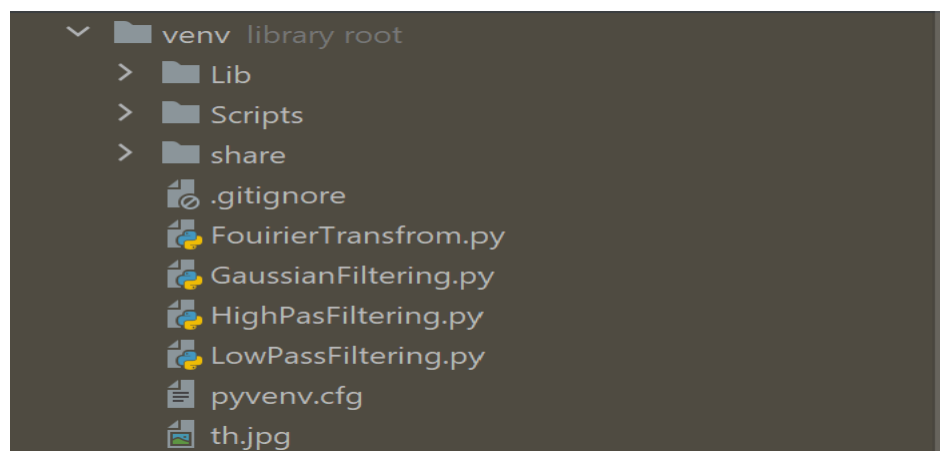
a. Buat *project* baru dengan nama *Domain Frekuensi*



b. Kemudian klik kanan pada folder *Domain Frekuensi – New – Python File*. Untuk membuat file baru.



c. File yang telah dibuat akan tampil seperti berikut



d. Selanjutnya ketikkan kode program pada masing-masing file yang telah dibuat. Seperti berikut :

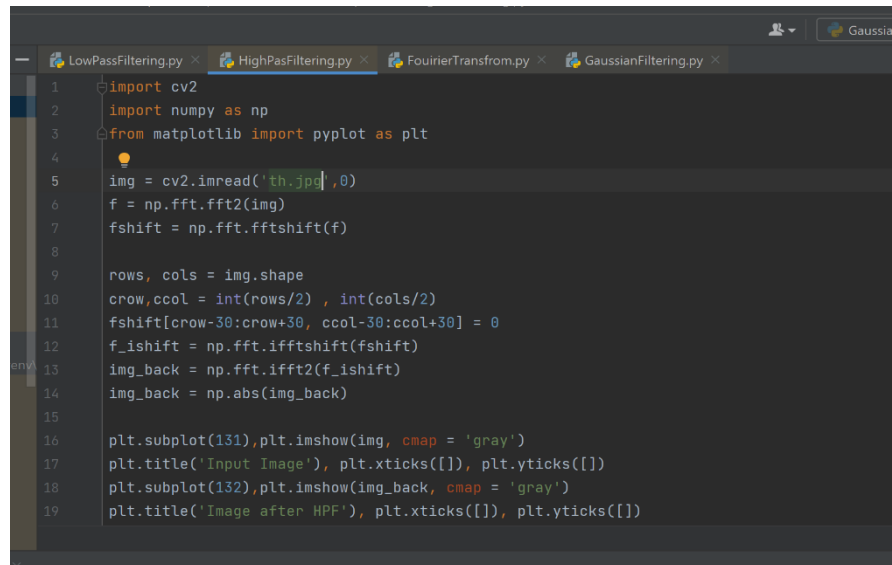
- *FourierTransform.py*

```
LowPassFiltering.py × HighPasFiltering.py × FoirierTransfrom.py × GaussianFiltering.py ×
1 import cv2
2 import numpy as np
3 from matplotlib import pyplot as plt
4
5 img = cv2.imread('th.jpg',0)
6 f = np.fft.fft2(img)
7 fshift = np.fft.fftshift(f)
8 magnitude_spectrum = 20*np.log(np.abs(fshift))
9
10 plt.subplot(121),plt.imshow(img, cmap = 'gray')
11 plt.title('Input Image'), plt.xticks([], plt.yticks([]))
12 plt.subplot(122),plt.imshow(magnitude_spectrum, cmap = 'gray')
13 plt.title('Magnitude Spectrum'), plt.xticks([], plt.yticks([]))
14 plt.show()
```

- *GaussianFiltering.py*

```
LowPassFiltering.py × HighPasFiltering.py × FoirierTransfrom.py × GaussianFiltering.py ×
1 import cv2
2 from matplotlib import pyplot as plt
3
4 img_path = "th.jpg"
5
6 img = cv2.imread(img_path)
7
8 blur_img = cv2.GaussianBlur(img, (9, 9), sigmaX=34, sigmaY=36)
9
10 plt.subplot(121),plt.imshow(img)
11 plt.title('Input Image'), plt.xticks([], plt.yticks([]))
12 plt.subplot(122),plt.imshow(blur_img)
13 plt.title('Image after GF'), plt.xticks([], plt.yticks([]))
14 plt.show()
```

- *HighPassFiltering.py*

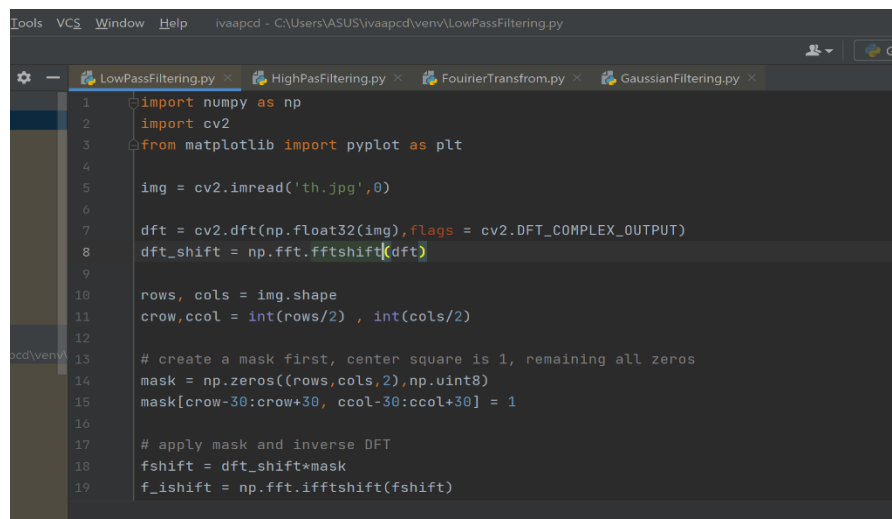


```

1  import cv2
2  import numpy as np
3  from matplotlib import pyplot as plt
4
5  img = cv2.imread('th.jpg',0)
6  f = np.fft.fft2(img)
7  fshift = np.fft.fftshift(f)
8
9  rows, cols = img.shape
10  crow,ccol = int(rows/2) , int(cols/2)
11  fshift[crow-30:crow+30, ccol-30:ccol+30] = 0
12  f_ishift = np.fft.ifftshift(fshift)
13  img_back = np.fft.ifft2(f_ishift)
14  img_back = np.abs(img_back)
15
16  plt.subplot(131),plt.imshow(img, cmap = 'gray')
17  plt.title('Input Image'), plt.xticks([], plt.yticks([]))
18  plt.subplot(132),plt.imshow(img_back, cmap = 'gray')
19  plt.title('Image after HPF'), plt.xticks([], plt.yticks([]))

```

- *LowPassFiltering.py*



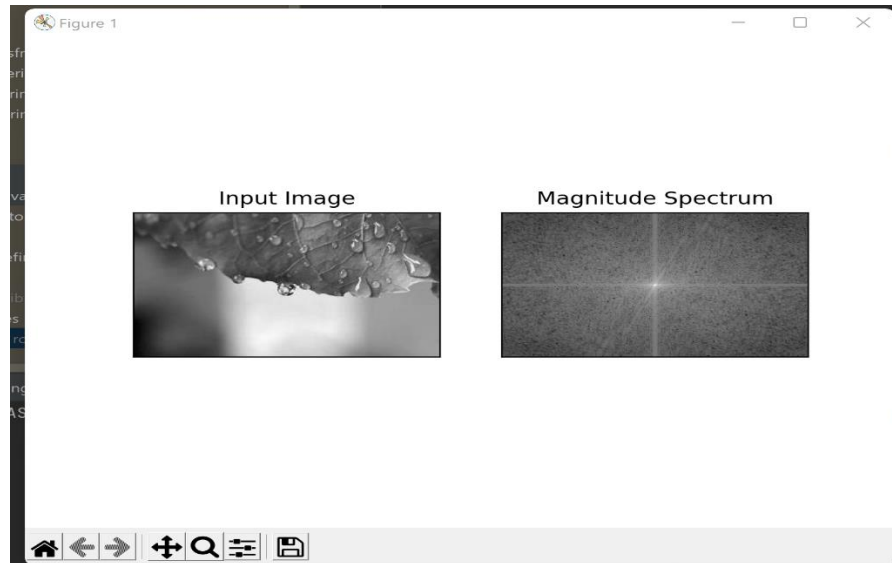
```

1  import numpy as np
2  import cv2
3  from matplotlib import pyplot as plt
4
5  img = cv2.imread('th.jpg',0)
6
7  dft = cv2.dft(np.float32(img),flags = cv2.DFT_COMPLEX_OUTPUT)
8  dft_shift = np.fft.fftshift(dft)
9
10 rows, cols = img.shape
11 crow,ccol = int(rows/2) , int(cols/2)
12
13 # create a mask first, center square is 1, remaining all zeros
14 mask = np.zeros((rows,cols,2),np.uint8)
15 mask[crow-30:crow+30, ccol-30:ccol+30] = 1
16
17 # apply mask and inverse DFT
18 fshift = dft_shift*mask
19 f_ishift = np.fft.ifftshift(fshift)

```

e. Kemudian Hasil dari program tersebut akan tampil seperti berikut :

- *FourierTransform.py*



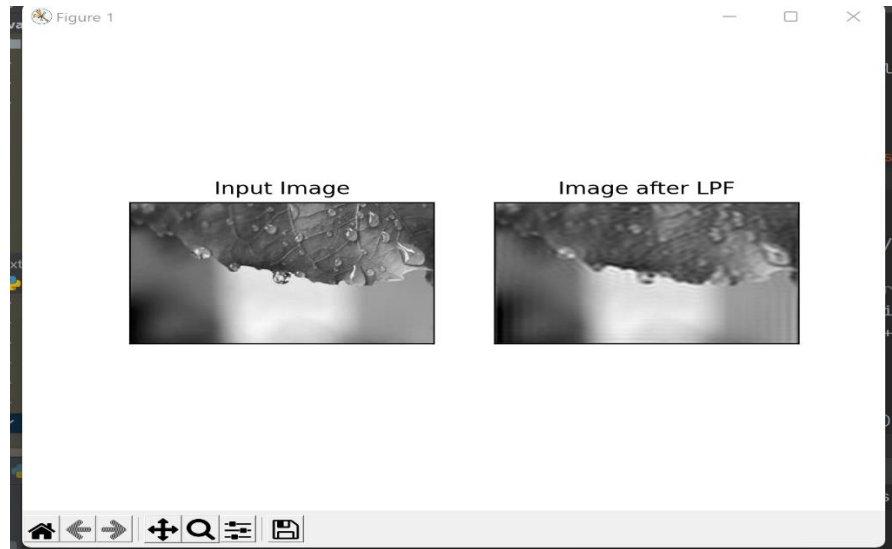
- *GaussianFiltering.py*



- *HighPassFiltering.py*



- *LowPassFiltering.py*



C. Analisis

Berdasarkan percobaan diatas dapat dianalisiskan bahwa untuk membuat program tersebut harus menginstal library python terlebih dahulu di antaranya opencv, matplotlib, dan library numpy agar program bisa berjalan. Setelah library berhasil terinstall copy gambar yang akan dijadikan percobaan. Terdapat kode `img = cv2.imread('th.jpg',0)` fungsinya untuk membaca gambar `fshift = np.fft.fftshift(f)` yang Menggeser suku frekuensi nol ke tengah larik. Untuk input dua dimensi, tukar kuadran pertama dan ketiga, dan kuadran kedua dan keempat. Dimana kode tersebut diatur untuk memberikan blurring pada gambar di two-dimensional. Sedangkan berikutnya, yaitu kode `20*np.log(np.abs(fshift))` merupakan pengaturan dimana diberikan paramete yang bisa digunakan pada fourier transformation. Berikutnya, tinggal mengatur citra gray pada gambar asli dan gambar hasil fourier transform yang akan ditampilkan dengan menggunakan subplot.

Berikutnya, untuk gaussian filtering diberikan variabel `blur_img` untuk mengatur blur pada gambar. Kode tersebut lengkapnya adalah `blur_img = cv2.GaussianBlur(img, (9, 9), sigmaX=34, sigmaY=36)`. Lalu, berikutnya mirip dengan kode sebelumnya yaitu pengaturan `plt.subplot` untuk mengatur subplot penampilan gambar. Olehnya hasil percobaan ini akan menampilkan gambar hasil yang berwarna dan blur Kemudian, untuk high pass filter disini juga diberikan kode yang sama seperti fourier transform yaitu pengaturan untuk blurring image two-dimensional dengan kode `f = np.fft.fft2(img)` dan `fshift = np.fft.fftshift(f)`.

D. Link Github

<https://github.com/SittiNurlailiQofifa/Sitti-Nurlaili-Qofifa-F55120107.git>