

Modul VII
Pengolahan citra digital
Domain Frekuensi

I. TUJUAN

1. Mahasiswa mampu mempelajari domain frekuensi.
2. Mahasiswa mampu memahami domain frekuensi
3. Mahasiswa mampu mengimplementasikan domain frekuensi.

II. ALAT DAN BAHAN

1. Laptop
2. *Pycharm*
3. *Python*
4. Citra gambar

III. TEORI DASAR

Kualitas citra merupakan hal yang paling krusial dalam semua bidang yang berbasis gambar. Citra yang berkualitas tinggi akan memudahkan pembacanya dalam memahami informasi yang terkandung didalamnya, sehingga mengurangi kesalahan yang mungkin terjadi. Berbicara mengenai kualitas citra, akan merujuk pada proses pengolahan citra, dimana pengolahan citra merupakan suatu proses yang dilakukan dengan masukan berupa citra dan hasilnya juga berupa citra dengan tujuan untuk meningkatkan kualitas dari citra tersebut. Salah satu metode yang dapat digunakan dalam proses perbaikan citra adalah Super Resolusi, Super Resolusi merupakan suatu teknik yang digunakan untuk membangun citra beresolusi tinggi dari sekumpulan citra yang memiliki resolusi rendah.

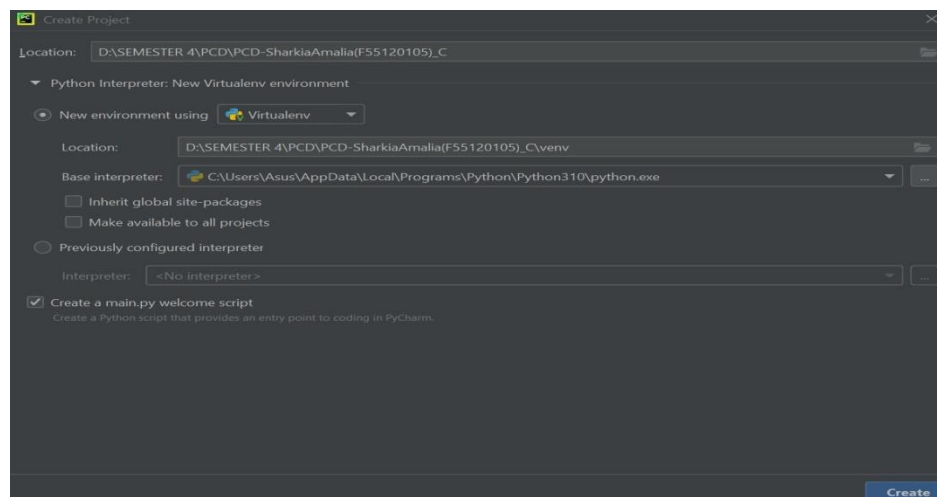
Teknik Super Resolusi mempunyai dua langkah utama, yaitu registrasi citra dan rekonstruksi citra. Registrasi merupakan langkah terpenting dalam teknik Super Resolusi karena registrasi citra menentukan

baik buruknya hasil dari perbaikan citra menggunakan Super Resolusi. Registrasi citra dapat diartikan sebagai proses untuk mendapatkan nilai pergeseran diantara citra beresolusi rendah yang melibatkan dua citra atau lebih yang memiliki objek sama namun diperoleh pada waktu, sudut pandang, dan sensor yang berbeda. Diterapkan salah satu metode dari registrasi citra yaitu metode *Power Cepstrum* yang bekerja dalam domain frekuensi. *Power Cepstrum* dapat digunakan untuk mendeteksi parameter rotasi dan translasi yang lebih efisien dan cukup akurat. Metode ini diperkenalkan pertama kali oleh Bogert et al, dimana awalnya metode ini digunakan untuk menganalisis data yang mengandung *echoes*.

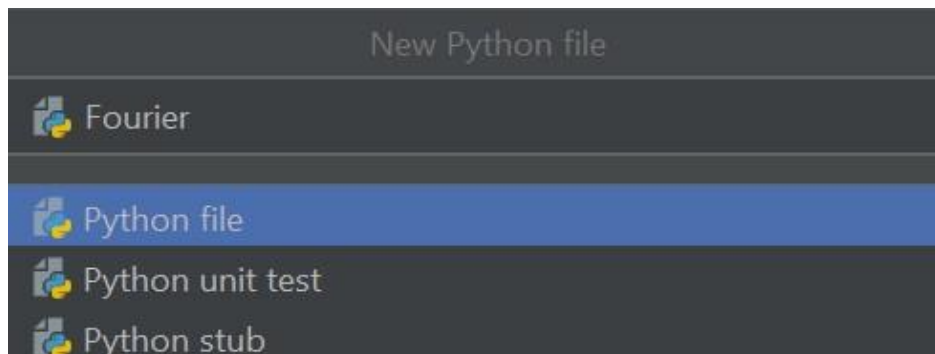
Power Cepstrum diperoleh dari logaritma spectrum sinyal yang kemudian ditransformasi menggunakan Transformasi *Fourier*. Metode ini mendapatkan nilai pergeseran dengan cara memproses nilai intensitas kedua citra referensi dan citra uji. Tujuan dari penelitian ini adalah membangun sebuah program yang dapat mendeteksi besar pergeseran diantara dua citra menggunakan metode *Power Cepstrum* dengan asumsi tidak ada beda rotasi diantara kedua citra. Program yang akan dibangun diharapkan dapat berkontribusi untuk menentukan pergeseran titik piksel citra dalam teknik Super Resolusi, sehingga bermanfaat dalam proses perbaikan citra.

IV. LANGKAH KERJA

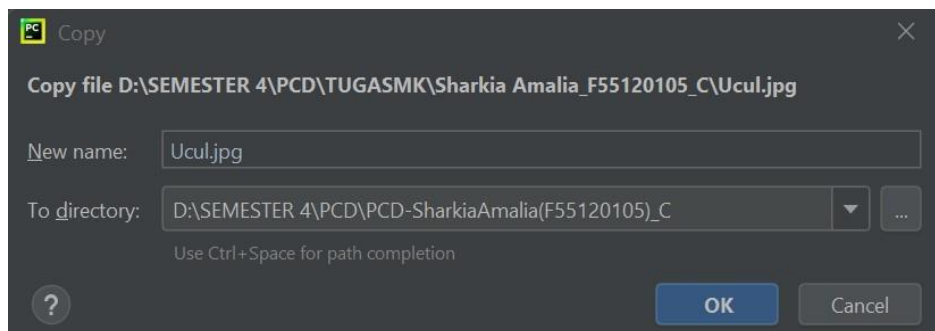
1. Membuka aplikasi *pycharm*, lalu membuat *project* baru.



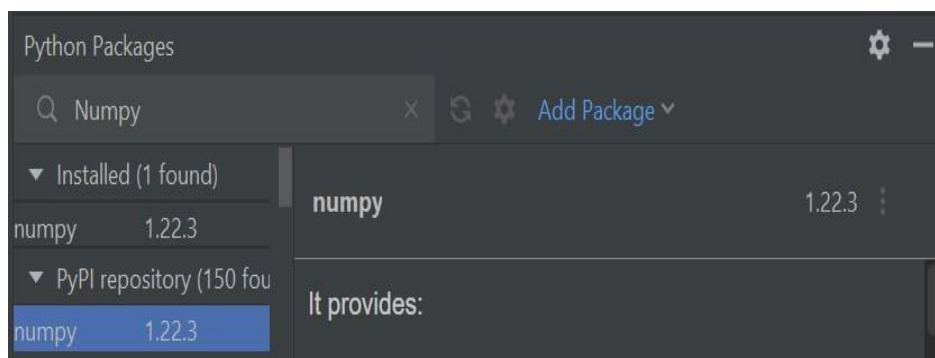
2. Membuat *python file* baru dengan nama “*fourier.py*”.



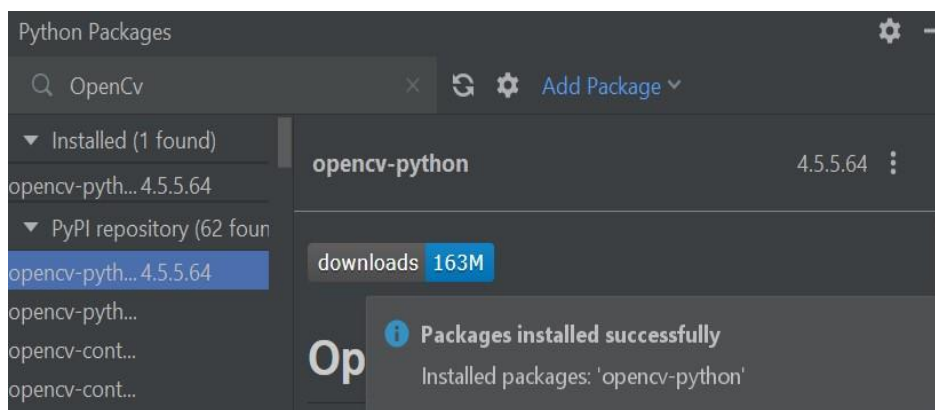
3. Menginputkan gambar ke dalam folder yang sama dengan *python file*.



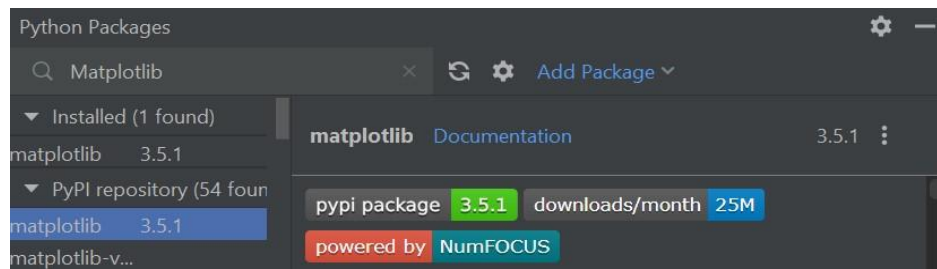
4. Menginstal *packages numpy*.



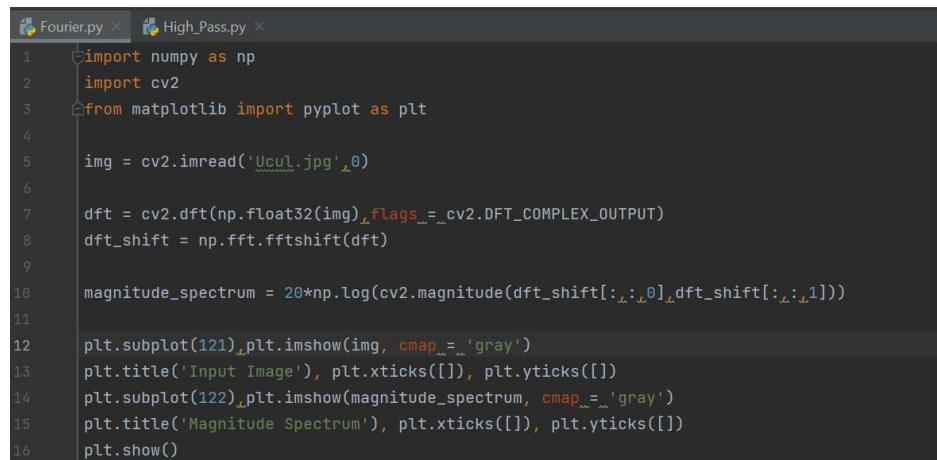
5. Menginstal *packages openCV*.



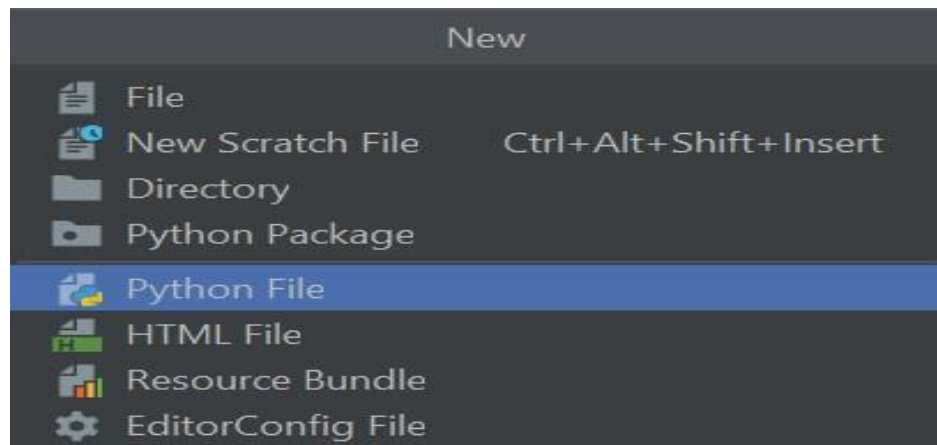
6. Menginstal *packages matplotlib*.



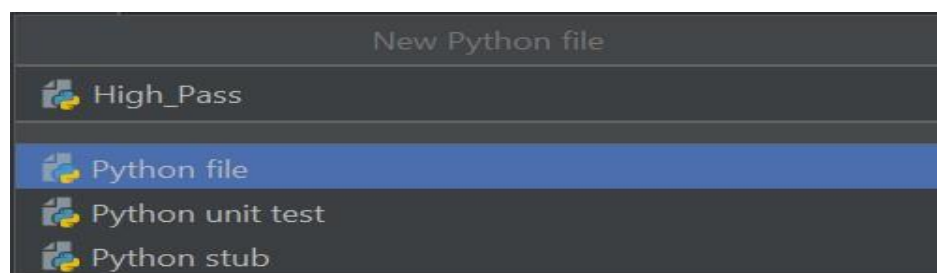
7. Menambahkan kode seperti berikut.



8. Membuat *file python* baru lagi dengan menekan “*file>new>python file*”.



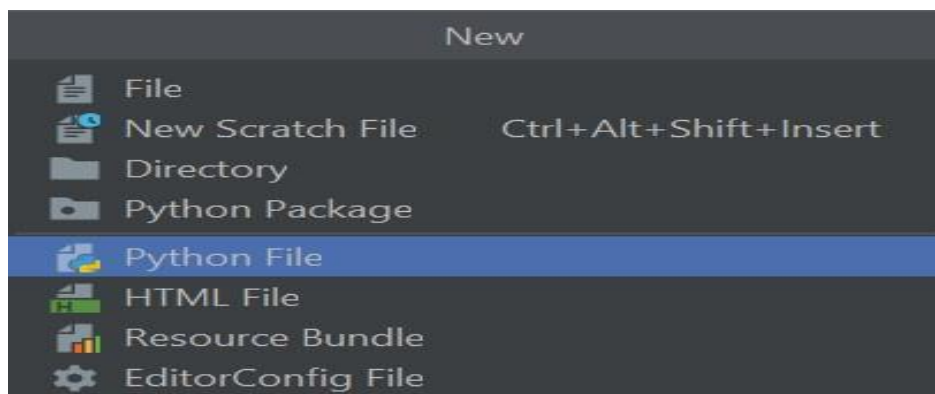
9. Memberi nama *file* dengan nama “*High_Pass.py*”



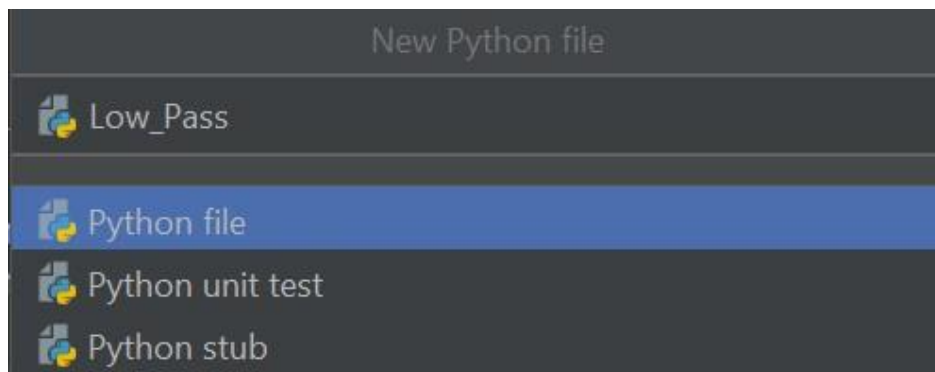
10. Menambahkan kode seperti berikut.

```
1 import cv2
2 import numpy as np
3 from matplotlib import pyplot as plt
4
5 img = cv2.imread('Ucu1.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 rows, cols = img.shape
11 crow, ccol = int(rows/2), int(cols/2)
12 fshift[crow-30:crow+30, ccol-30:ccol+30] = 0
13 f_ishift = np.fft.ifftshift(fshift)
14 img_back = np.fft.ifft2(f_ishift)
15 img_back = np.abs(img_back)
16
17 plt.subplot(131), plt.imshow(img, cmap='gray')
18 plt.title('Input Image'), plt.xticks([], plt.yticks([]))
19 plt.subplot(132), plt.imshow(img_back, cmap='gray')
20 plt.title('Image after HPF'), plt.xticks([], plt.yticks([]))
21 plt.subplot(133), plt.imshow(img_back)
22 plt.title('Result in JET'), plt.xticks([], plt.yticks([]))
23 plt.show()
```

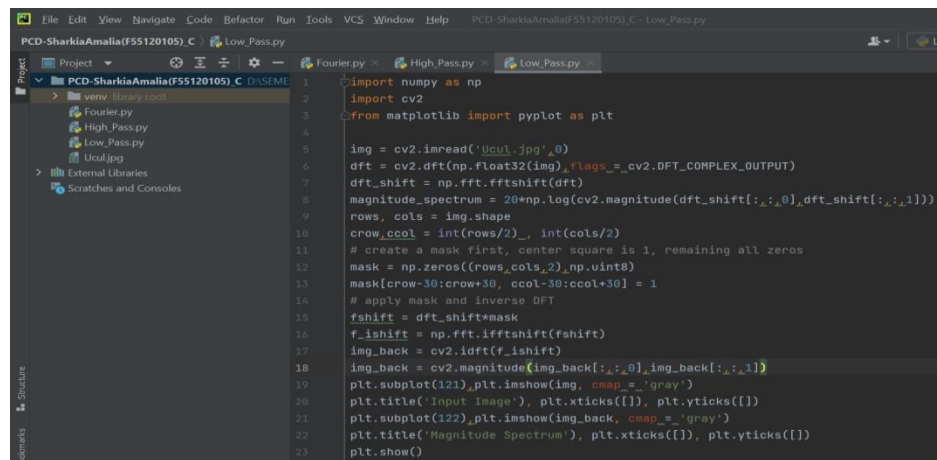
11. Membuat *file python* baru lagi dengan menekan “*file>new>python file*”



12. Meberi nama *file* dengan nama “*Low_Pass.py*”.



13. Menambahkan kode seperti berikut.

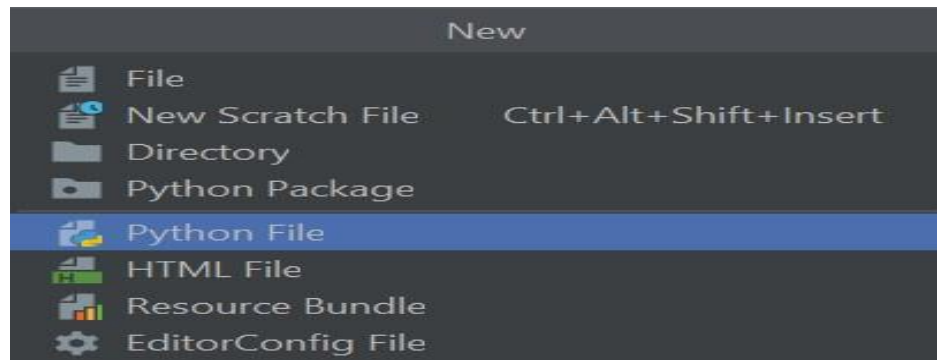


```

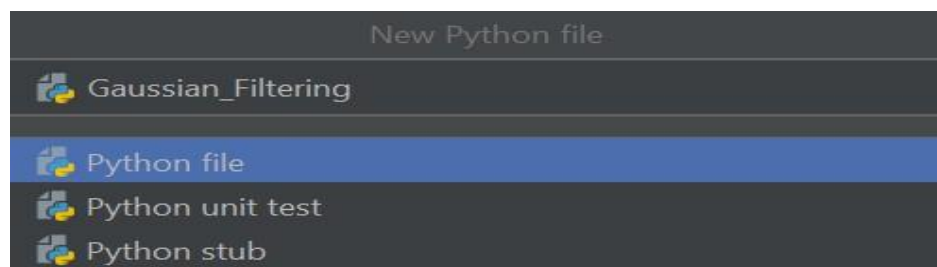
1 import numpy as np
2 import cv2
3 from matplotlib import pyplot as plt
4
5 img = cv2.imread('Ucu1.jpg', 0)
6 dft = cv2.dft(np.float32(img), flags=cv2.DFT_COMPLEX_OUTPUT)
7 dft_shift = np.fft.fftshift(dft)
8 magnitude_spectrum = 20*np.log(cv2.magnitude(dft_shift[:, :, 0], dft_shift[:, :, 1]))
9
10 rows, cols = img.shape
11 crows, ccols = int(rows/2), int(cols/2)
12 # create a mask first, center square is 1, remaining all zeros
13 mask = np.zeros((rows, cols, 2), np.uint8)
14 mask[crows-30:crows+30, ccols-30:ccols+30] = 1
15 # apply mask and inverse DFT
16 fshift = dft_shift*mask
17 f_ishift = np.fft.ifftshift(fshift)
18 img_back = cv2.idft(f_ishift)
19 img_back = cv2.magnitude(img_back[:, :, 0], img_back[:, :, 1])
20 plt.subplot(121), plt.imshow(img, cmap='gray')
21 plt.title('Input Image'), plt.xticks([], plt.yticks([]))
22 plt.subplot(122), plt.imshow(img_back, cmap='gray')
23 plt.title('Magnitude Spectrum'), plt.xticks([], plt.yticks([]))
24 plt.show()

```

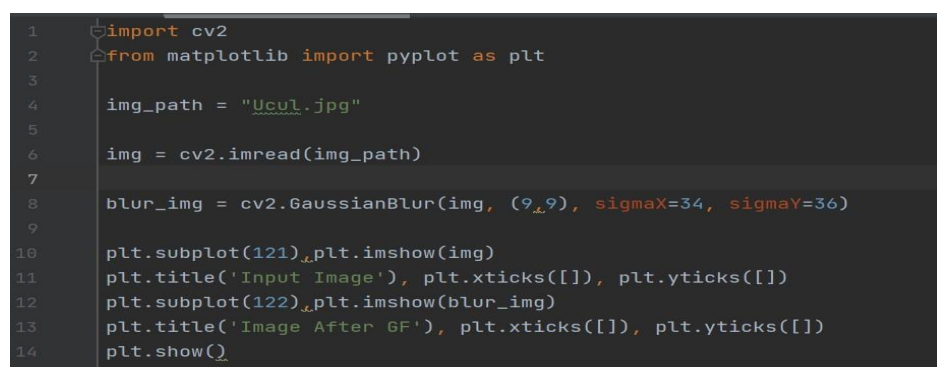
14. Membuat *file python* baru lagi dengan menekan “*file>new>python file*”



15. Meberi nama *file* dengan nama “*Gaussian_filtering.py*”.



16. Menambahkan kode seperti berikut.



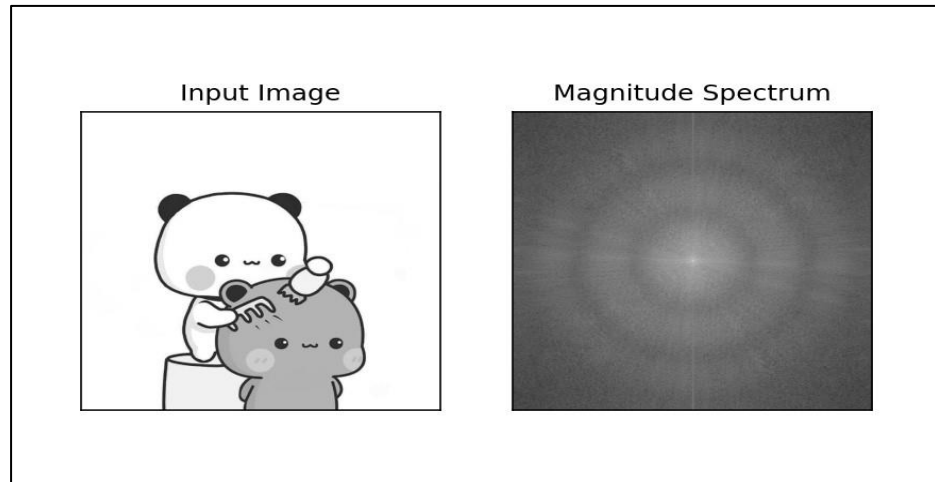
```

1 import cv2
2 from matplotlib import pyplot as plt
3
4 img_path = "Ucu1.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()

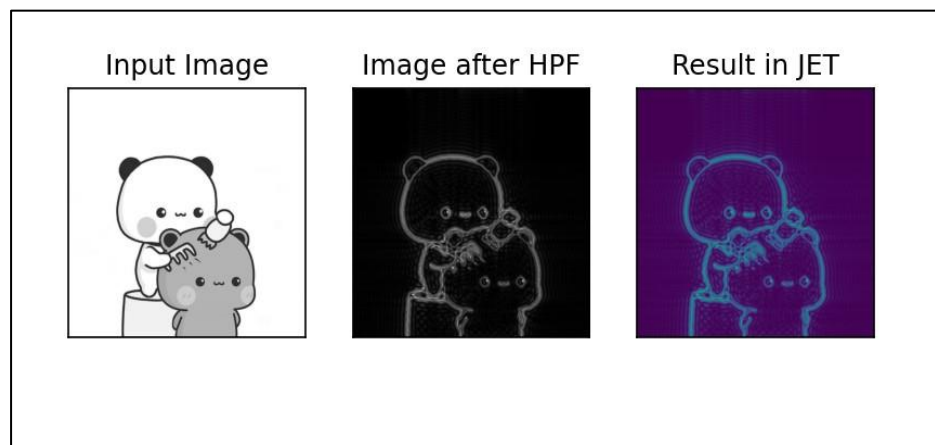
```

V. HASIL PERCOBAAN

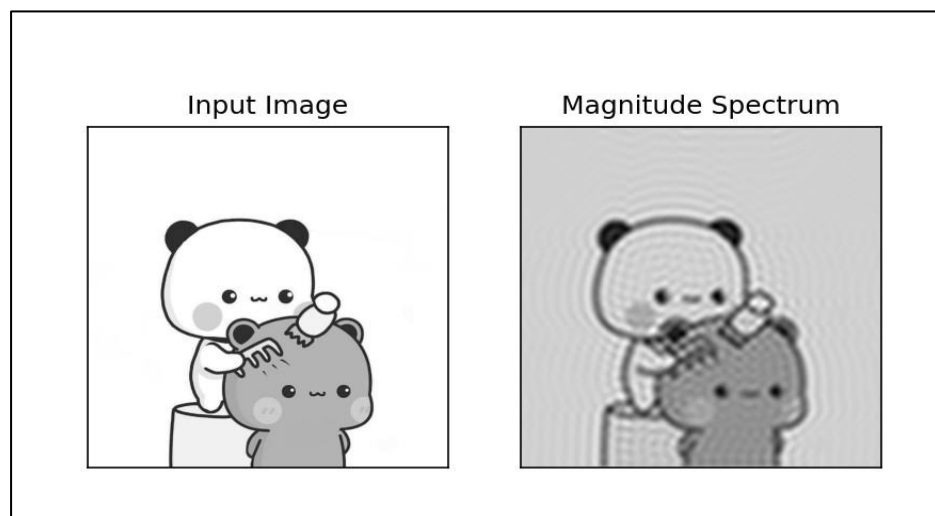
1. Hasil *Fourier*



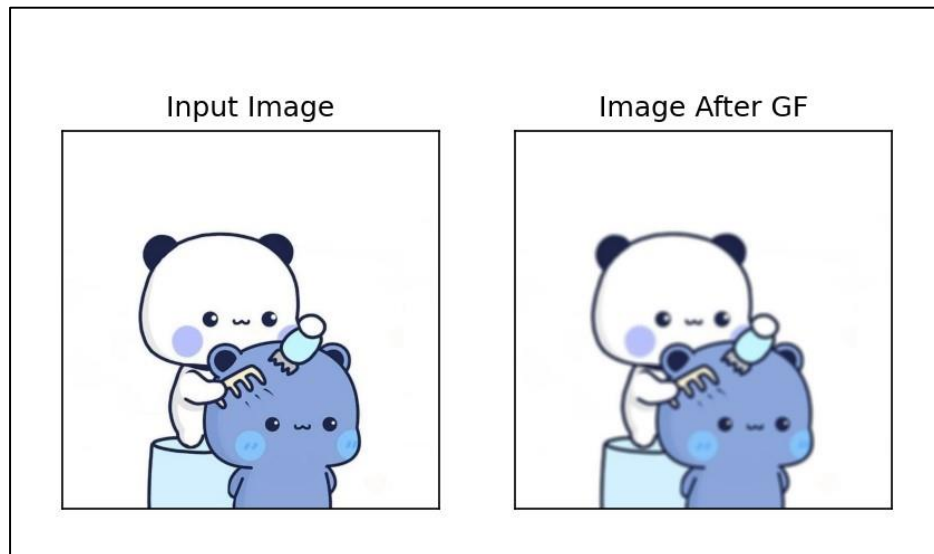
2. Hasil *High_Pass*



3. Hasil *Low_Pass*



4. Hasil *Gaussian*



VI. ANALISIS

Terdapat 3 domain frekuensi yaitu *fourier*, *high* dan *low pass* dan *gaussian*, yang dimana *fourier* adalah sebuah transformasi integral yang menyatakan-kembali sebuah fungsi dalam fungsi basis sinusoidal, yaitu sebuah fungsi sinusoidal penjumlahan atau integral dikalikan oleh beberapa koefisien ("amplitudo") yang fungsinya adalah digunakan untuk mengubah domain spasial pada citra pada citra menjadi domain frekuensi. Kemudian *high pass* adalah filter yang hanya melewatkan frekuensi tinggi. dan *low pas* adalah filter yang hanya melewatkan frekuensi rendah, aplikasi dari rangkaian ini adalah pada speaker untuk digunakan sebagai output frekuensi rendah atau *woofer*. Dan *Gaussian* dapat diartikan sebagai sebuah efek blur yang dihasilkan oleh sebuah fungsi gaussian. biasanya efek ini dihadirkan dalam beragam jenis *software editing*. fungsinya adalah untuk mereduksi noise yang muncul pada gambar serta meningkatkan efek blur pada foto.

Pada *fourier* digunakan *library numpy* yang berfungsi untuk proses komputasi numerik. Kemudian *library cv2* yang dimana *openCv* adalah sebuah *library open source* yang dikembangkan oleh intel yang fokus untuk menyederhanakan programing terkait citra digital. Dan *library matplotlib* yang berfungsi untuk melakukan visualisasi data seperti membuat *plot*

grafik untuk satu sumbu atau lebih. Yang dimana citra dibaca dengan memasukkan kode `"img = cv2.imread('ucul.jpg',0)"` kemudian dimasukkan kode untuk memasukkan perintah *magnitude_spectrum* yaitu `"magnitude_spectrum = 20*np.log(cv2.magnitude(dft_shift[:, :, 0], dft_shift[:, :, 1]))"`.

Kemudian pada *high pass* sama seperti *fourier*, digunakan library yang sama namun pada kode selanjutnya yaitu `"f = np.fft.fft2(img)"` yaitu digunakan untuk perhitungan Discrete Fourier Transform atau DFT. Kemudian `"magnitude_spectrum = 20*np.log(np.abs(fshift))"` yaitu perintah *magnitude_spectrum* untuk *high pass* kemudian dimasukkan kode untuk proses *high pass filter* dan ditampilkan dengan perintah *subplot* dan *imshow* dan pada *lowpass filter* memiliki proses *filtering* yang berbeda.

Dan yang terakhir *Gaussian* hanya menggunakan library *cv2* dan *matplotlib* kemudian terdapat kode `"blur_img = cv2.GaussianBlur(img, (9, 9), sigmaX=34, sigmaY=36)"` sebagai kode untuk proses *Gaussian filtering*nya

VII. KESIMPULAN

Dari percobaan disimpulkann terdapat beberapa domain frekuensi pada pengolahan citra digital seperti contohnya yaitu *fourier*, *high pass*, *low pass*, dan *Gaussian filtering* yang masing-masing memiliki fungsi tersendiri.