

TUGAS PENGOLAHAN CITRA DIGITAL

DOMAIN FREKUENSI



Disusun Oleh :

RESTI AMANDA

F 551 20 121

C

PROGRAM STUDI S1 TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

FAKULTAS TEKNIK

UNIVERSITAS TADULAKO

2022

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.

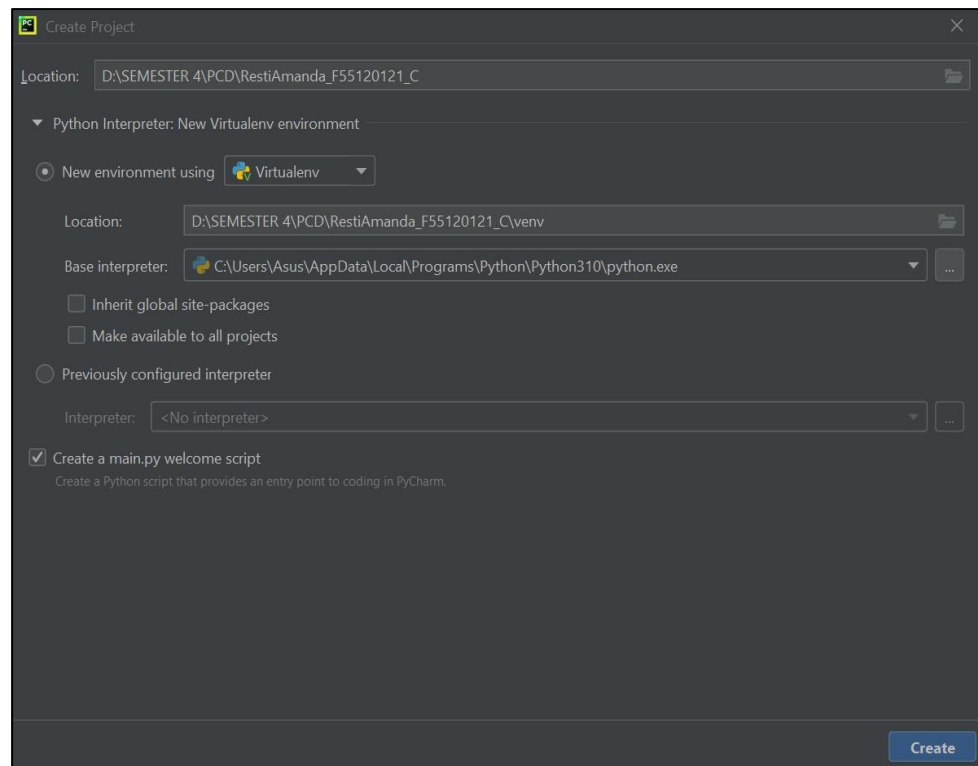
Teknik Super Resolusi mempunyai dua langkah utama, yaitu registrasi citra dan rekonstruksi citra. Registrasi merupakan langkah terpenting dalam teknik Super Resolusi karena registrasi citra menentukan baik buruknya hasil dari perbaikan citra menggunakan Super Resolusi. Registrasi citra dapat diartikan 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. Dalam Tugas Akhir ini, akan 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.

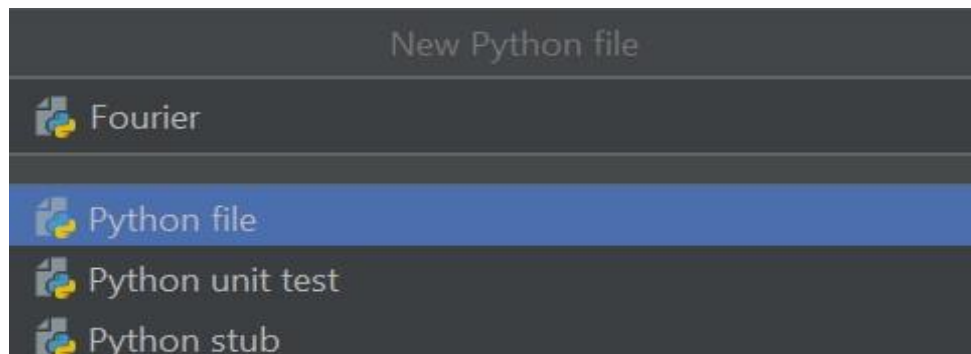
Power Cepstrum diperoleh dari logaritma *spectrum* sinyal yang kemudian ditransformasi menggunakan Transformasi *Fourier*. Metode ini mendapatkan nilai pergeseran dengan cara memproses nilai intensitas kedua citra referensi dan citra uji. Tujuan dari penelitian ini adalah membangun sebuah program yang 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

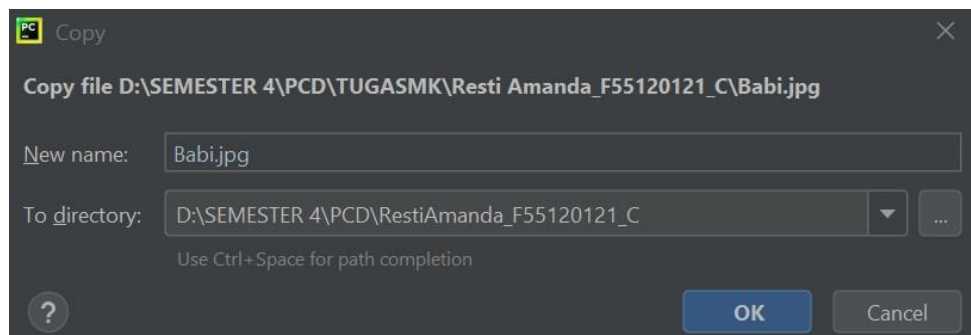
1. Membuka aplikasi *pycharm*, Kemudian membuat *project* baru.



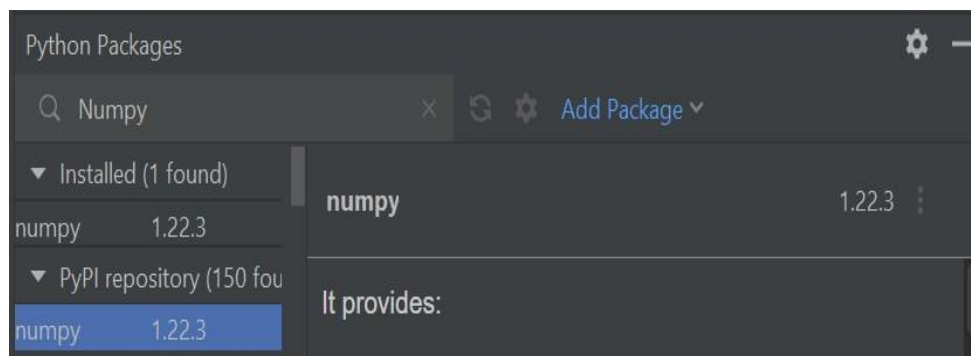
2. Membuat *python file* baru dengan nama “*fourier.py*”.



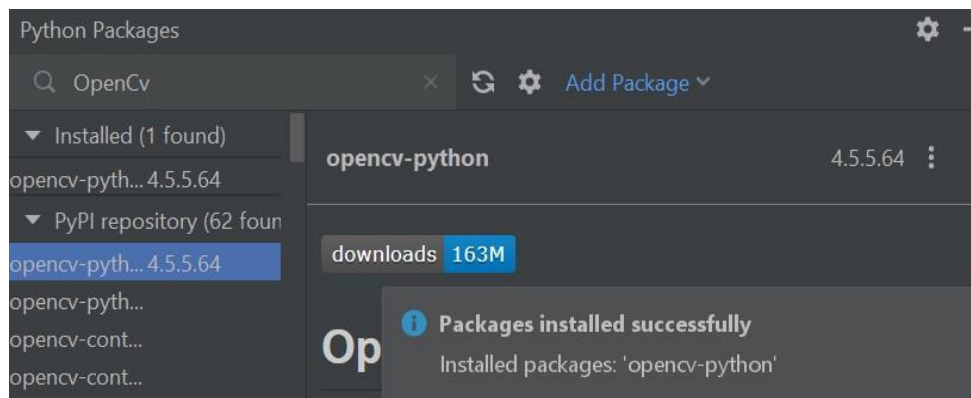
3. Menginputkan gambar ke dalam folder yang sama dengan *python file*.



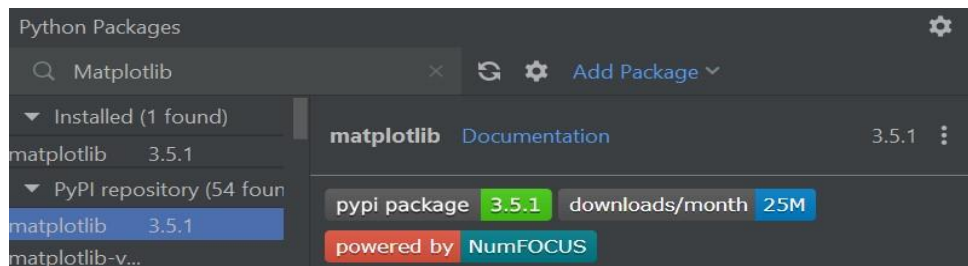
4. Menginstal *packages numpy*.



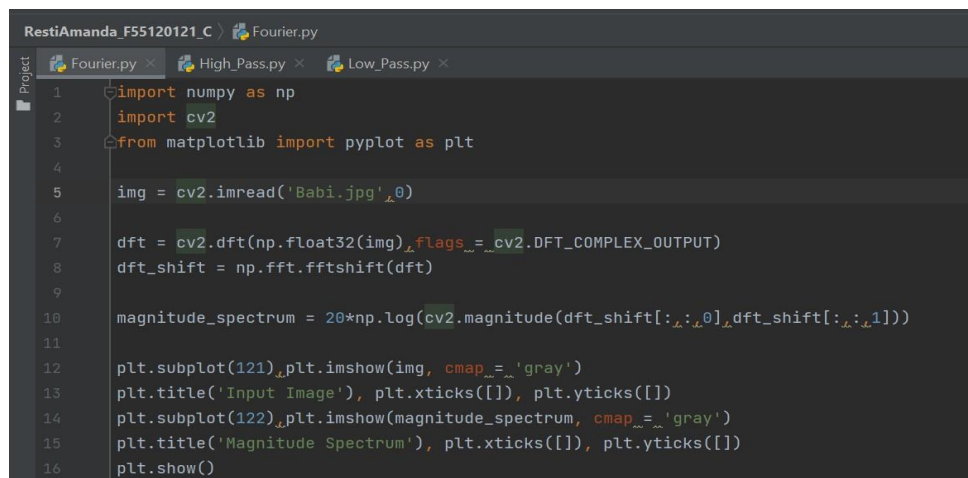
5. Menginstal *packages openCV*.



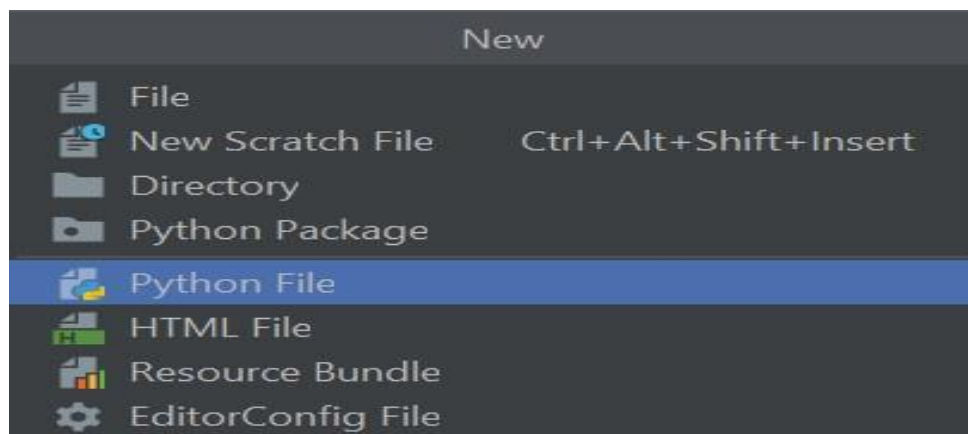
6. Menginstal *packages matplotlib*.



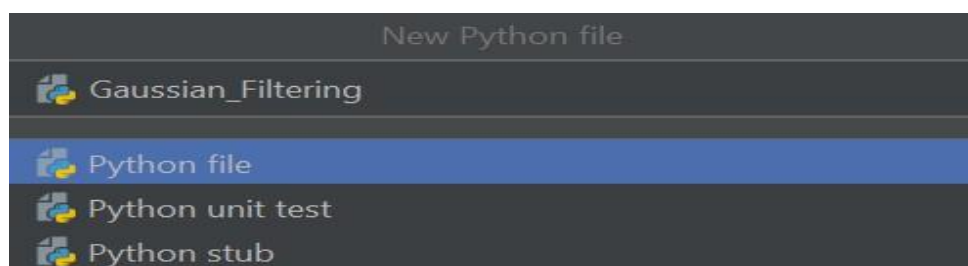
7. Menambahkan kode seperti berikut pada “*fourier.py*”.



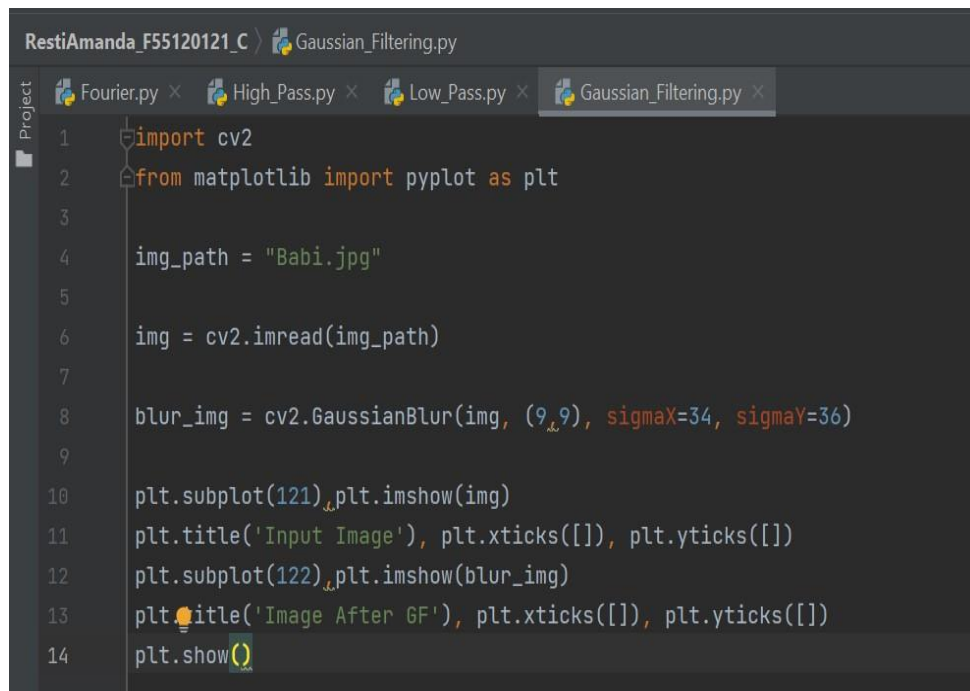
8. Membuat *file python* baru lagi dengan menekan “*file>new>python file*”.



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

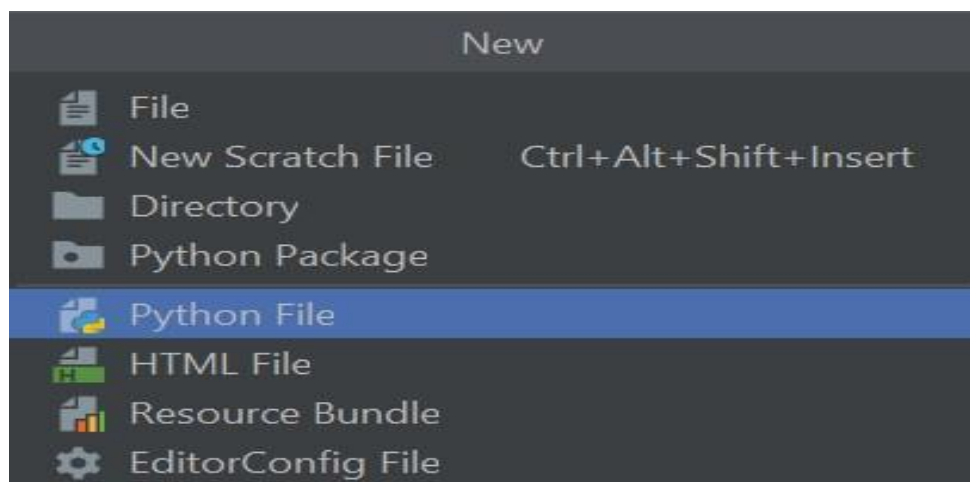


10. Menambahkan kode seperti berikut pada “*Gaussian_Filtering.py*”

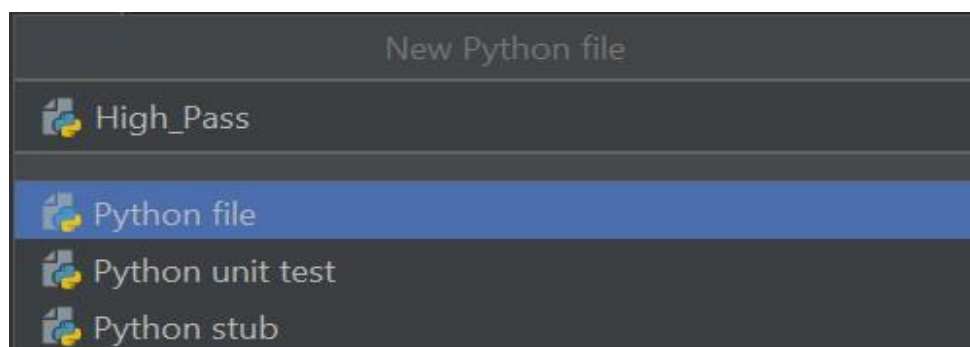


```
RestiAmanda_F55120121_C > Gaussian_Filtering.py
Project
1  import cv2
2  from matplotlib import pyplot as plt
3
4  img_path = "Babi.jpg"
5
6  img = cv2.imread(img_path)
7
8  blur_img = cv2.GaussianBlur(img, (9,9), sigmaX=34, sigmaY=36)
9
10 plt.subplot(121),plt.imshow(img)
11 plt.title('Input Image'), 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()
```

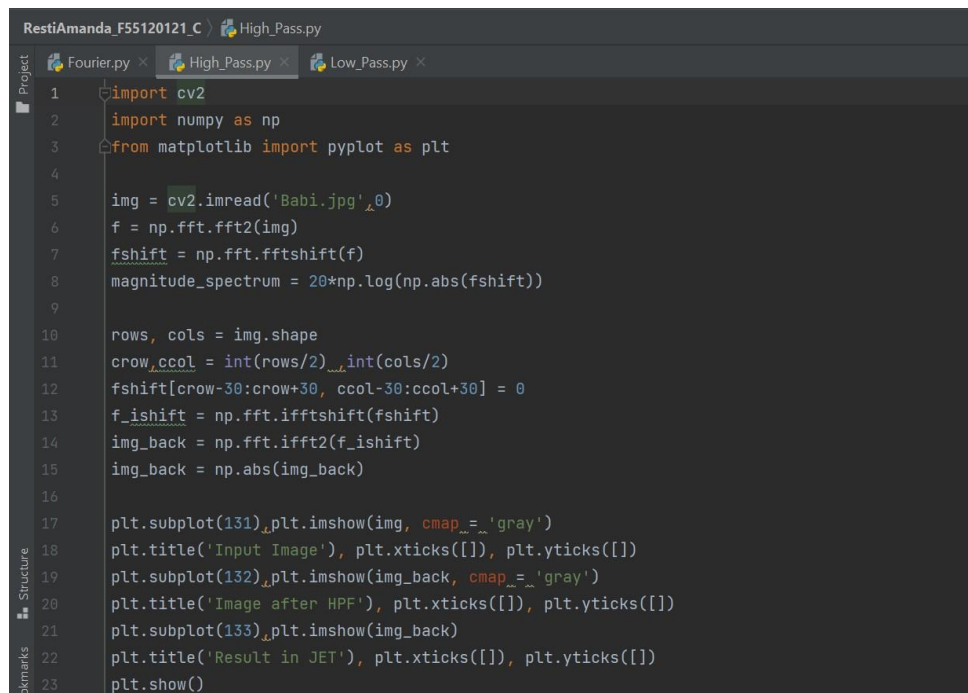
11. Membuat *file python* baru lagi dengan menekan “*file>new>python file*”.



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

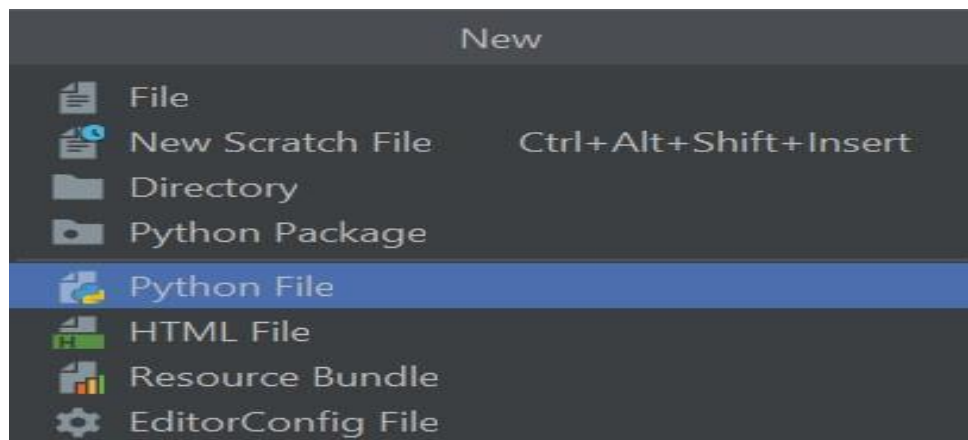


13. Menambahkan kode seperti berikut pada “*High_Pass.py*”

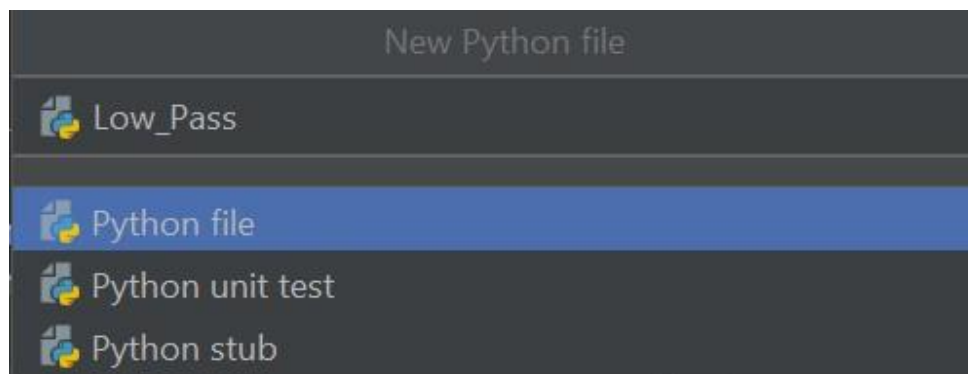


```
RestiAmanda_F55120121_C > High_Pass.py
1 import cv2
2 import numpy as np
3 from matplotlib import pyplot as plt
4
5 img = cv2.imread('Babi.jpg', 0)
6 f = np.fft.fft2(img)
7 fshift = np.fft.fftshift(f)
8 magnitude_spectrum = 20*np.log(np.abs(fshift))
9
10 rows, cols = img.shape
11 crow, ccol = int(rows/2), 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()
```

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



15. Memberi nama *file* dengan nama “*Low_Pass.py*”.

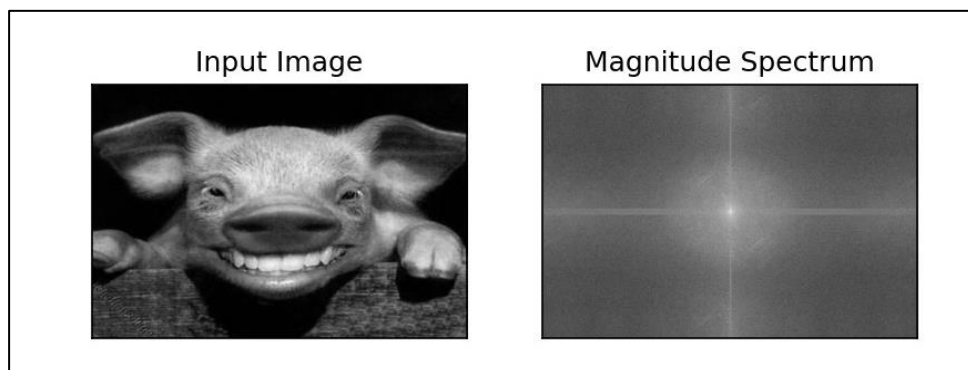


16. Menambahkan kode seperti berikut pada “*Low_Pass.py*”.

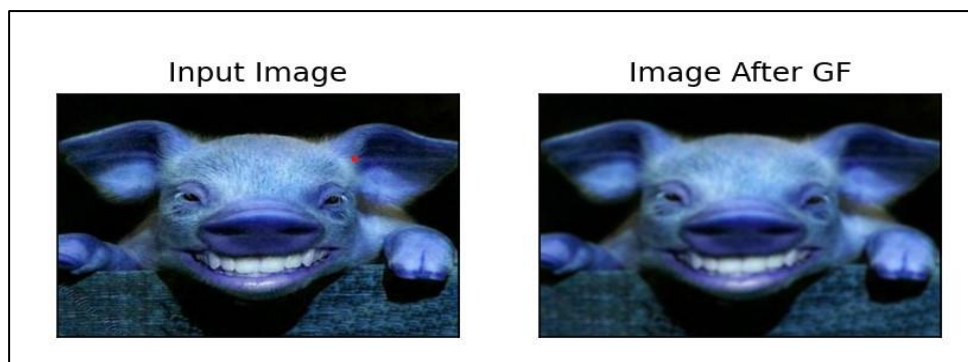
```
RestiAmanda_F55120121_C > Low_Pass.py
Fourier.py x High_Pass.py x Low_Pass.py x
1 import numpy as np
2 import cv2
3 from matplotlib import pyplot as plt
4
5 img = cv2.imread('Babi.jpg', 0)
6 dft = cv2.dft(np.float32(img), flags=cv2.DFT_COMPLEX_OUTPUT)
7 dft_shift = np.fft.fftshift(dft)
8 magnitude_spectrum = 20*np.log(cv2.magnitude(dft_shift[:, :, 0], dft_shift[:, :, 1]))
9 rows, cols = img.shape
10 crow, ccol = int(rows/2), int(cols/2)
11 # create a mask first, center square is 1, remaining all zeros
12 mask = np.zeros((rows, cols), np.uint8)
13 mask[crow-30:crow+30, ccol-30:ccol+30] = 1
14 # apply mask and inverse DFT
15 fshift = dft_shift*mask
16 f_ishift = np.fft.ifftshift(fshift)
17 img_back = cv2.idft(f_ishift)
18 img_back = cv2.magnitude(img_back[:, :, 0], img_back[:, :, 1])
19 plt.subplot(121), plt.imshow(img, cmap=gray)
20 plt.title('Input Image'), plt.xticks([], plt.yticks([]))
21 plt.subplot(122), plt.imshow(img_back, cmap=gray)
22 plt.title('Magnitude Spectrum'), plt.xticks([], plt.yticks([]))
23 plt.show()
```

V. HASIL PERCOBAAN

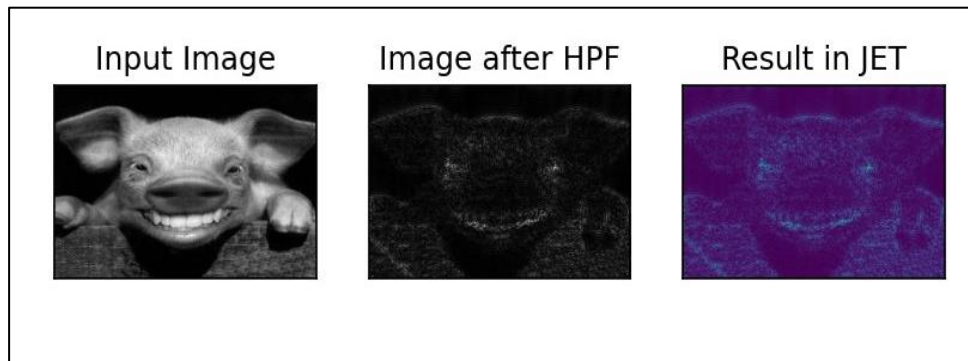
1. Hasil *Fourier*



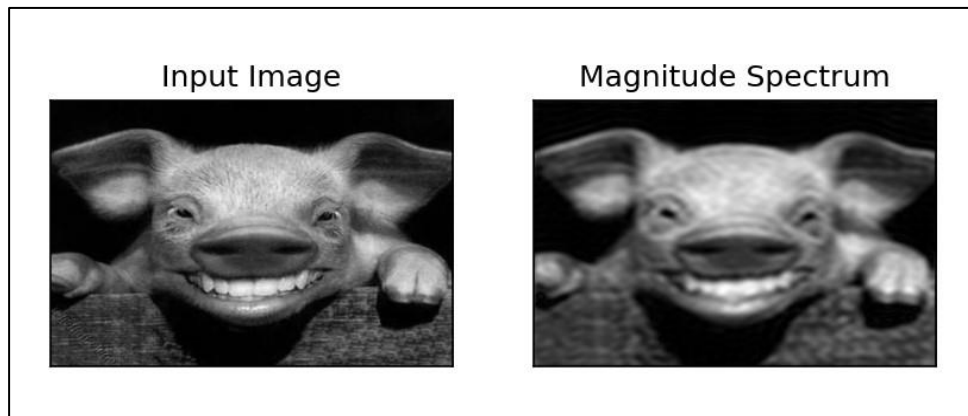
2. Hasil *Gaussian Filtering*



3. Hasil *High Pass*



4. Hasil *Low Pass*



VI. ANALISIS

Pada percobaan ini dapat dianalisis bahwa dalam domain frekuensi terbagi menjadi beberapa operasi yaitu transformasi *Fourier*, *Gaussian Filtering*, *Low Pass*, dan juga *High Pass*. Hal utama yang dilakukan jika ingin membuat suatu program pada IDE *PyCharm* ialah menginstall *packages* yang dibutuhkan setelah itu mengimport *library* nya. Adapun *library* yang diimportkan pada percobaan ini ialah *Numpy* yang berfungsi dalam proses pengolahan data *numeric*, *Cv2* atau *OpenCv-Python* yang berfokus dalam penyederhanaan program terkait dengan citra digital. Dan yang terakhir ialah *matplotlib* yang berfungsi dalam melakukan visualisasi data.

Pada '*Fourier.py*' terdapat kode program `"img = cv2.imread('Babi.jpg', 0)"` yang berfungsi untuk membaca file gambar. `"magnitude_spectrum = 20*np.log(cv2.magnitude(dft_shift[:, :, 0], dft_shift[:, :, 1]))"` yang dimana

merupakan perintah dari *magnitude spectrum*. Selanjutnya pada file *'High_Pass.py'* terdapat *"f = np.fft.fft2(img)"* yang dimana digunakan dalam melakukan perhitungan *Discrete Fourier Transform (DFT)*. *"plt.subplot(132), plt.imshow(img, cmap = 'gray')"* yang dimana terdapat perintah *subplot* yang dimana *subplot* berfungsi agar hasilnya nanti akan tampil secara terplot dalam satu *window*. 1 ialah jumlah baris, 3 merupakan jumlah kolomnya, dan 2 ialah urutannya.

VII. KESIMPULAN

Pada percobaan ini dapat disimpulkan bahwa Domain frekuensi adalah metode yang digunakan untuk menganalisis data. Dalam domain frekuensi terbagi menjadi *fourier, gaussian filtering, high pass* dan *low pass*.