

**PENGOLAHAN CITRA DIGITAL**  
**TUGAS KE-4**

Tanggal Pengumpulan : 25 Mei 2017



Nama Mahasiswa	: Rizky Ajie Aprilianto
NIM / Rombel	: 5301414037 / 02
Nama Dosen	: Dr. Hari Wibawanto, M.T. Kuntoro Adi Nugroho, S.T., M.Eng.

**PENDIDIKAN TEKNIK ELEKTRO JURUSAN TEKNIK ELEKTRO**  
**FAKULTAS TEKNIK UNIVERSITAS NEGERI SEMARANG**  
**SEMARANG**

**2017**

## TUGAS 4

Buatlah *filter image* menggunakan *High Pass Filter* (HPF), *Low Pass Filter* (LPF) dan *Histogram image*!

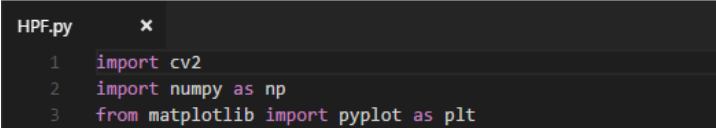
## JAWABAN

### 1. *High Pass Filter* (HPF)

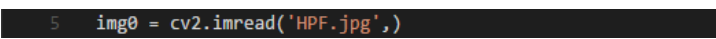
*High Pass Filter* (HPF) bisa digunakan untuk membuat gambar terlihat tajam. HPF bekerja sama dengan *Low Pass Filter*, hanya saja yang berbeda ada pada konvolusi kernel. Salah satu penggunaan HPF adalah menemukan tepi dari sebuah gambar. HPF digunakan untuk menemukan tepi gambar. *Edge Detection* (deteksi garis tepi) biasanya digunakan untuk mendeteksi tepi sebuah skema (sobel), penggunaan dasar deteksi tepi dan Laplacian. Berikut *source code High Pass Filter* (HPF):

```
HPF.py      x
1  import cv2
2  import numpy as np
3  from matplotlib import pyplot as plt
4
5  img0 = cv2.imread('HPF.jpg',)
6
7  gray = cv2.cvtColor(img0, cv2.COLOR_BGR2GRAY)
8
9  img = cv2.GaussianBlur(gray,(3,3),0)
10
11  laplacian = cv2.Laplacian(img, cv2.CV_64F)
12  sobelx = cv2.Sobel(img,cv2.CV_64F,1,0,ksize=5)
13  sobely = cv2.Sobel(img,cv2.CV_64F,0,1,ksize=5)
14
15  plt.subplot(2,2,1), plt.imshow(img,cmap = 'gray')
16  plt.title('Original'), plt.xticks([], plt.yticks([]))
17  plt.subplot(2,2,2), plt.imshow(laplacian,cmap = 'gray')
18  plt.title('Laplacian'), plt.xticks([], plt.yticks([]))
19  plt.subplot(2,2,3), plt.imshow(sobelx, cmap = 'gray')
20  plt.title('Sobel X'), plt.xticks([], plt.yticks([]))
21  plt.subplot(2,2,4), plt.imshow(sobely,cmap = 'gray')
22  plt.title('Sobel Y'), plt.xticks([], plt.yticks([]))
23
24  plt.show()
```

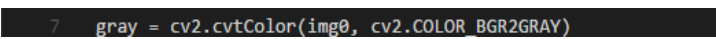
### Penjelasan Source Code:

a. 

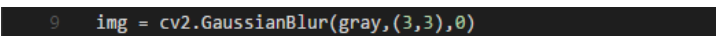
Perintah diatas digunakan untuk mengimpor *library* yang akan digunakan.

b. 

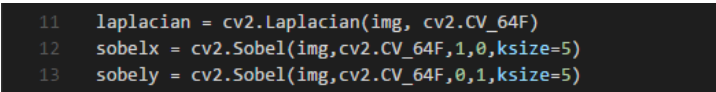
Perintah diatas digunakan untuk melakukan inisialisasi pada gambar.

c. 

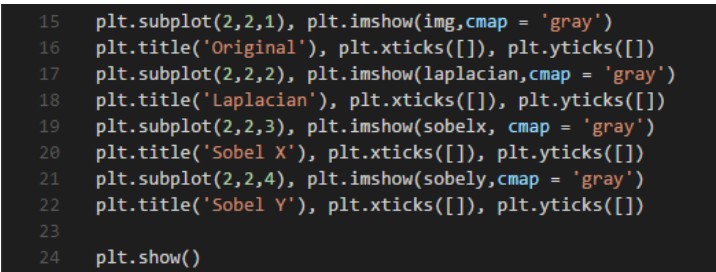
Perintah diatas digunakan untuk menginisialisasi gambar dan merubah gambar warna menjadi abu-abu.

d. 

Perintah diatas digunakan untuk menginisialisai gambar dan merubahnya menjadi efek kabur Gaussian, karena efek gaussian sangat baik digunakan untuk menghilangkan noise.

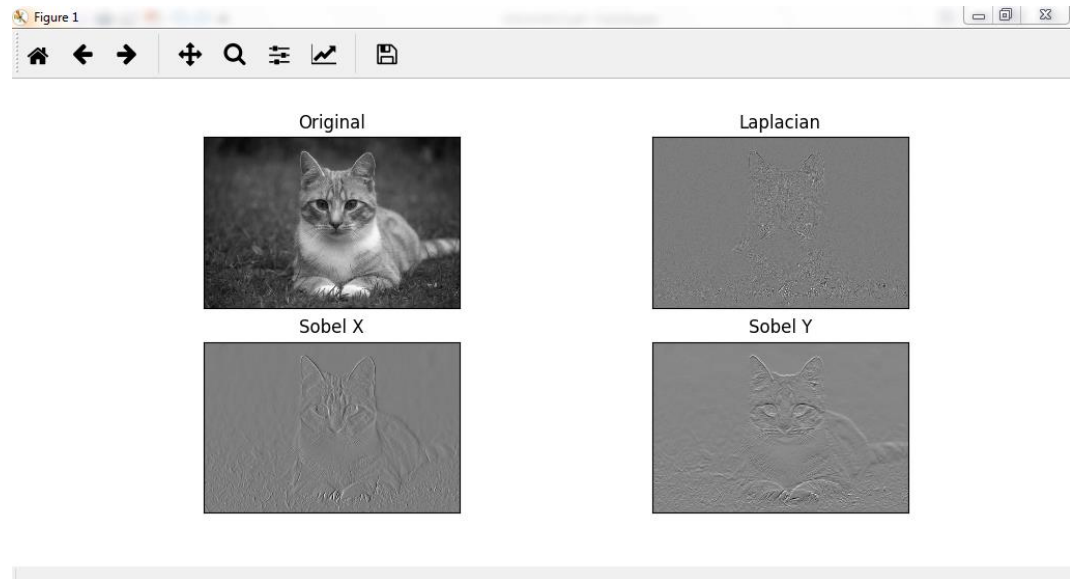
e. 

Perintah diatas digunakan untuk mengkonvolusi gambar dengan kernel yang tepat. Pada efek Laplacian, cv2.CV\_64F digunakan untuk meminta ukuran gambar tujuan. Sedangkan sobelx menginisialisasi gambar menjadi efek sobel x-dir, dan sobely gambar diinisialisasi menjadi efek sobel y-dir terhadap axes Y.

f. 

Perintah diatas digunakan menampilkan hasil gambar kedalam satu *figure*.

## Hasil *High Pass Filter* (HPF):



## 2. *Low Pass Filter* (LPF)

*Low Pass Filter* (LPF) bekerja dengan menghapus ruang derau frekuensi tinggi dari gambar digital. LPF digunakan untuk menggeser jendela penghubung antara efek *pixel* pertama dari gambar pada waktu yang sama, mengubah nilai dari suatu fungsi lokal region *pixel*. LPF digunakan juga untuk membantu menghapus noises, menghaburkan (*blurring*) gambar. Dalam *source code* berikut ini, LPF digunakan untuk mengaburkan gambar (*blurring*) dengan efek Average dan Gaussian.

```
LPF.py      x
1  import cv2
2  import numpy as np
3  from matplotlib import pyplot as plt
4
5  img = cv2.imread('LPF.jpg')
6
7  blur = cv2.blur(img,(5,5))
8  blur2 = cv2.GaussianBlur(img,(5,5),0)
9
10 plt.subplot(2,2,1),plt.imshow(img),plt.title('Gambar Asli')
11 plt.xticks([], plt.yticks([]))
12 plt.subplot(2,2,2),plt.imshow(blur),plt.title('Blured')
13 plt.xticks([], plt.yticks([]))
14 plt.subplot(2,2,3),plt.imshow(blur2),plt.title('Gaussian Blur')
15 plt.xticks([], plt.yticks([]))
16 plt.show()
```

### Penjelasan *Source Code*:

a.

```
LPF.py  x
1  import cv2
2  import numpy as np
3  from matplotlib import pyplot as plt
```

Perintah diatas digunakan untuk mengimpor *library* yang akan digunakan.

b.

```
5  img = cv2.imread('LPF.jpg')
```

Perintah diatas digunakan untuk melakukan inisialisasi pada gambar.

c.

```
7  blur = cv2.blur(img,(5,5))
```

Perintah diatas digunakan untuk mengganti setiap nilai *pixel* didalam sebuah gambar dengan rata-rata nilai dari *pixel* tetanggannya, dan mencakup dirinya sendiri. Sehingga didapatkan efek gambar kabur. Fungsi ini juga dikenal dengan Averaging blurred, dimana gambar digulung dengan filter ternormal kotak. Penggunaan (5,5) adalah kernel lebar dan tinggi 5x5. Ini digunakan untuk merepresentasikan bentuk dan ukuran dari sekitarnya sebagai sample kemudian mengkalkulasi dengan rata-rata.

d.

```
8  blur2 = cv2.GaussianBlur(img,(5,5),0)
```

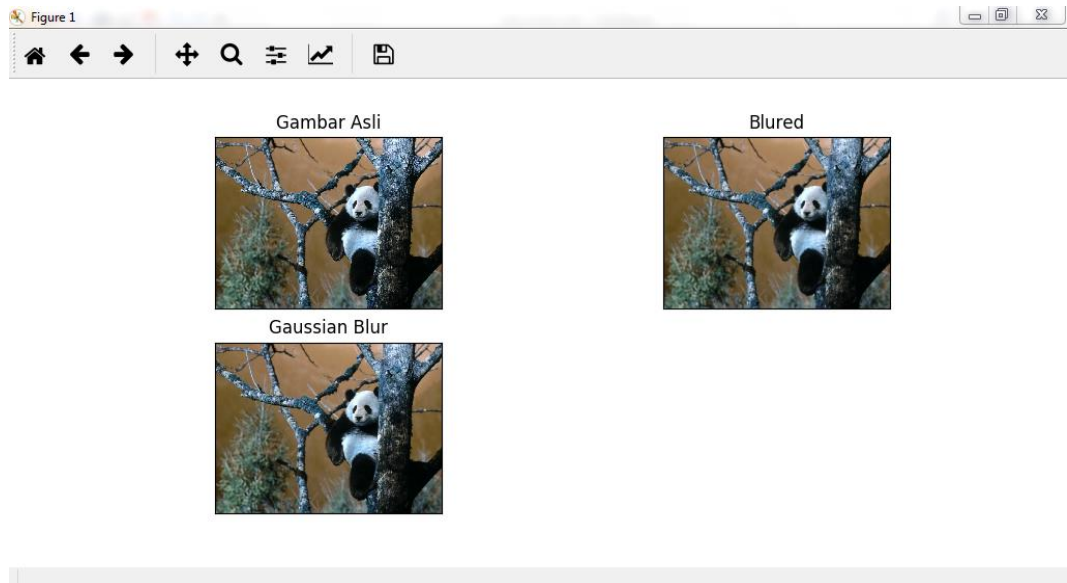
Salah satu contoh aplikasi *Low Pass Filter* adalah efek Gaussian bluring. Menggunakan fungsi cv2.GaussianBlur(). Dalam teori Gaussian, setiap titik dari gambar akan terhitung tidak-nol, sehingga seluruh gambar harus dihitung setiap pixelnya. (5,5) adalah ukuran kernel lebar dan tinggi.

e.

```
10 plt.subplot(2,2,1),plt.imshow(img),plt.title('Gambar Asli')
11 plt.xticks([], plt.yticks([]))
12 plt.subplot(2,2,2),plt.imshow(blur),plt.title('Blured')
13 plt.xticks([], plt.yticks([]))
14 plt.subplot(2,2,3),plt.imshow(blur2),plt.title('Gaussian Blur')
15 plt.xticks([], plt.yticks([]))
16 plt.show()
```

Perintah diatas digunakan untuk menampilkan hasil gambar kedalam satu *figure*.

### Hasil *Low Pass Filter* (LPF) dengan efek Average dan Gaussian:



### 3. Histogram

Histogram digunakan untuk merepresentasikan *graphical* dari distribusi intensitas gambar. Digunakan pula untuk mengkuantisasi nomor *pixel* tiap nilai intensitas. Berikut *source code* histogram:

```
Histogram.py x
1  import cv2
2  import numpy as np
3  from matplotlib import pyplot as plt
4
5  gray_img = cv2.imread('Histogram.jpg', cv2.IMREAD_GRAYSCALE)
6  cv2.imshow('Histogram', gray_img)
7  hist = cv2.calcHist([gray_img],[0],None,[256],[0,256])
8
9  plt.hist(gray_img.ravel(),256,[0,256])
10 plt.title('Histogram for gray scale picture')
11 plt.show()
12
13 while True:
14     k = cv2.waitKey(0) & 0xFF
15     if k == 27 : break
16 cv2.destroyAllWindows()
```

## Penjelasan Source Code:

a. Histogram.py ✕

```
1 import cv2
2 import numpy as np
3 from matplotlib import pyplot as plt
```

Perintah diatas digunakan untuk mengimpor *library* yang akan digunakan.

b.

```
5 gray_img = cv2.imread('Histogram.jpg', cv2.IMREAD_GRAYSCALE)
6 cv2.imshow('Histogram', gray_img)
```

Perintah diatas digunakan untuk membaca gambar dan merubahnya menjadi gambar *grayscale*. Kemudian menampilkannya kedalam gambar.

c.

```
7 hist = cv2.calcHist([gray_img],[0],None,[256],[0,256])
```

cv2.calcHist() adalah fungsi untuk histogram. [gray\_img], gambar yang digunakan adalah type uint8 atau folat32. [0] digunakan karena input gambar adalah grayscale, jika menggunakan gambar warna dapat menggunakan [0], [1] atau [2] untuk mengkalkulasi histogram channel biru, hijau dan merah, berturut-turut. None, adalah mask image yang digunakan untuk menemukan histogram gambar penuh. [256] adalah histSize untuk skala penuh. [0,256] adalah *range* normal.

d.

```
9 plt.hist(gray_img.ravel(),256,[0,256])
10 plt.title('Histogram for gray scale picture')
11 plt.show()
```

Perintah diatas digunakan untuk menampilkan gambar kedalam *figure* dengan nama “Histogram for gray scale picture”.

e.

```
13 while True:
14     k = cv2.waitKey(0) & 0xFF
15     if k == 27 : break
16     cv2.destroyAllWindows()
```

Perintah diatas digunakan untuk menutup *figure* jika tombol ESC ditekan.

## Hasil Histogram:

