

PENGOLAHAN CITRA DIGITAL



| | |
|----------------|--|
| Nama | : Khonif Nur Fitriyah |
| NIM / Rombel | : 5301414043 / 02 |
| Dosen Pengampu | : Dr. Hari Wibawanto, M.T. Kuntoro Adi Nugroho, S.T, M.Eng. |

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

TUGAS 2

Buatlah filter image menggunakan Low Pass Filter dan High Pass Filter kemudian buatlah Histogram dari hasil filter tersebut!

JAWAB:

Berdasarkan efeknya image filtering dibedakan menjadi dua kelompok, yaitu:

1. High Pass Filter (HPF)

Filter ini digunakan untuk mempertahankan titik yang berbeda dengan titik-titik tetangganya (*proses deteksi tepi*), suatu bentuk filter yang mengambil data pada frekwensi tinggi dan membuang data pada frekwensi rendah. HPF membuat gambar terlihat tajam. Selain itu HPF dapat bekerja sama dengan Low Pass Filter (LPF), tetapi konvolusi kernelnya berbeda. Salah satu penggunaan HPF adalah menemukan tepi dari sebuah gambar.

Saya menggunakan HPF untuk menemukan tepi gambar. Edge detection (deteksi garis tepi) biasanya digunakan untuk mendeteksi tepi sebuah skema (sobel) penggunaan dasar deteksi tepi dan Laplacian. File python edge detection ini saya simpan dengan nama **filter2.py**. Berikut source code nya:

```

1  import cv2
2  import numpy as np
3  from matplotlib import pyplot as plt
4
5  # loading image
6  img0 = cv2.imread('Nurul.jpg',)
7
8  # mengubah ke skala keabuan
9  gray = cv2.cvtColor(img0, cv2.COLOR_BGR2GRAY)
10
11 # menghapus derau
12 img = cv2.GaussianBlur(gray,(3,3),0)
13
14 # konvolusi dengan proper kernels
15 laplacian = cv2.Laplacian(img, cv2.CV_64F)
16 sobelx = cv2.Sobel(img,cv2.CV_64F,1,0,ksize=5)
17 sobely = cv2.Sobel(img,cv2.CV_64F,0,1,ksize=5)
18
19 plt.subplot(2,2,1), plt.imshow(img,cmap = 'gray')
20 plt.title('Original'), plt.xticks([], plt.yticks([]))
21 plt.subplot(2,2,2), plt.imshow(laplacian,cmap = 'gray')
22 plt.title('Laplacian'), plt.xticks([], plt.yticks([]))
23 plt.subplot(2,2,3), plt.imshow(sobelx, cmap = 'gray')
24 plt.title('Sobel X'), plt.xticks([], plt.yticks([]))
25 plt.subplot(2,2,4), plt.imshow(sobely,cmap = 'gray')
26 plt.title('Sobel Y'), plt.xticks([], plt.yticks([]))
27 |
28 plt.show()

```

🚧 Penjelasan:

```

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

```

Untuk mengimport modul atau library yang akan digunakan

```

5  # loading image
6  img0 = cv2.imread('Nurul.jpg',)
7

```

Untuk menginisialisasi gambar

```

8  # mengubah ke skala keabuan
9  gray = cv2.cvtColor(img0, cv2.COLOR_BGR2GRAY)
10

```

Untuk mengubah gambar ke skala keabuan

```

15 img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
16 # mengkonvolusikan gambar dengan kernel Gaussian

```

Untuk menghilangkan derau dan mengubah gambar menjadi efek kabur Gaussian.

```

14 # konvolusi dengan proper kernels
15 laplacian = cv2.Laplacian(img, cv2.CV_64F)
16 sobelx = cv2.Sobel(img, cv2.CV_64F, 1, 0, ksize=5)
17 sobely = cv2.Sobel(img, cv2.CV_64F, 0, 1, ksize=5)
18

```

Untuk mengkonvolusikan 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 di inialisasi menjadi efek sobel y-dir terhadap axes Y

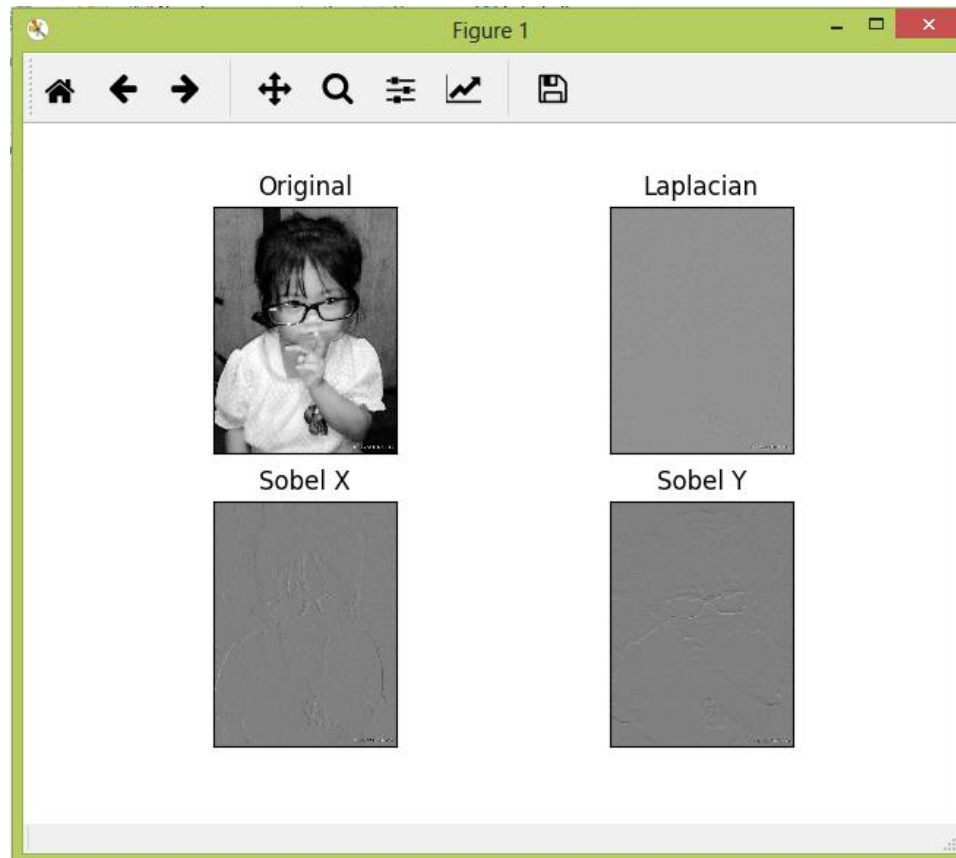
```

18
19 plt.subplot(2,2,1), plt.imshow(img, cmap = 'gray')
20 plt.title('Original'), plt.xticks([]), plt.yticks([])
21 plt.subplot(2,2,2), plt.imshow(laplacian, cmap = 'gray')
22 plt.title('Laplacian'), plt.xticks([]), plt.yticks([])
23 plt.subplot(2,2,3), plt.imshow(sobelx, cmap = 'gray')
24 plt.title('Sobel X'), plt.xticks([]), plt.yticks([])
25 plt.subplot(2,2,4), plt.imshow(sobely, cmap = 'gray')
26 plt.title('Sobel Y'), plt.xticks([]), plt.yticks([])
27
28 plt.show()

```

Untuk menampilkan gambar dalam satu bingkai (figure).

🚦 **Hasil:**



2. Low Pass Filter (LPF)

Filter ini digunakan untuk membuang titik yang berbeda dengan titik-titik tetangganya (*proses reduksi noise*), yaitu suatu bentuk filter yang mengambil data pada frekuensi rendah dan membuang data pada frekuensi tinggi. LPF dapat digunakan untuk menghapus derau, dan menghamburkan gambar (blurring).

Saya menggunakan LPF untuk menghamburkan gambar (blurring) dengan efek average dan Gaussian. Program yang saya gunakan adalah openCV, dengan nama file **filter1.py**. Berikut source code-nya:

```

1  import cv2
2  import numpy as np
3  from matplotlib import pyplot as plt
4  #untuk mengimport modul atau library
5
6  img = cv2.imread('Kamboja.jpg')
7  #untuk inisialisasi gambar
8
9  blur = cv2.blur(img,(5,5))
10 #memberikan efek blur
11
12 blur2 = cv2.GaussianBlur(img,(5,5),0)
13 #memberikan efek gaussian blur
14
15 plt.subplot(2,2,1),plt.imshow(img),plt.title('Gambar Asli')
16 plt.xticks([], plt.yticks([]))
17 plt.subplot(2,2,2),plt.imshow(blur),plt.title('Blured')
18 plt.xticks([], plt.yticks([]))
19 plt.subplot(2,2,3),plt.imshow(blur2),plt.title('Gaussian Blur')
20 plt.xticks([], plt.yticks([]))
21 plt.show()
22 #menampilkan gambar dalam satu bingkai

```

🚩 Penjelasan:

```

1  import cv2
2  import numpy as np
3  from matplotlib import pyplot as plt
4  #untuk mengimport modul atau library

```

Untuk mengimport modul atau library yang akan digunakan

```

6  img = cv2.imread('Kamboja.jpg')
7  #untuk inisialisasi gambar

```

Untuk menginisialisasi gambar

```

9  blur = cv2.blur(img,(5,5))
10 #memberikan efek blur

```

Untuk memberikan efek blur dengan mengganti setiap nilai pixel di dalam sebuah gambar dengan rata-rata nilai dari pixel tetangganya dan mencakup dirinya sendiri. Sehingga didapatkan efek gambar kabur. Fungsi ini juga dikenal dengan Averaging blured, 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 mengkalculasi dengan rata-rata.

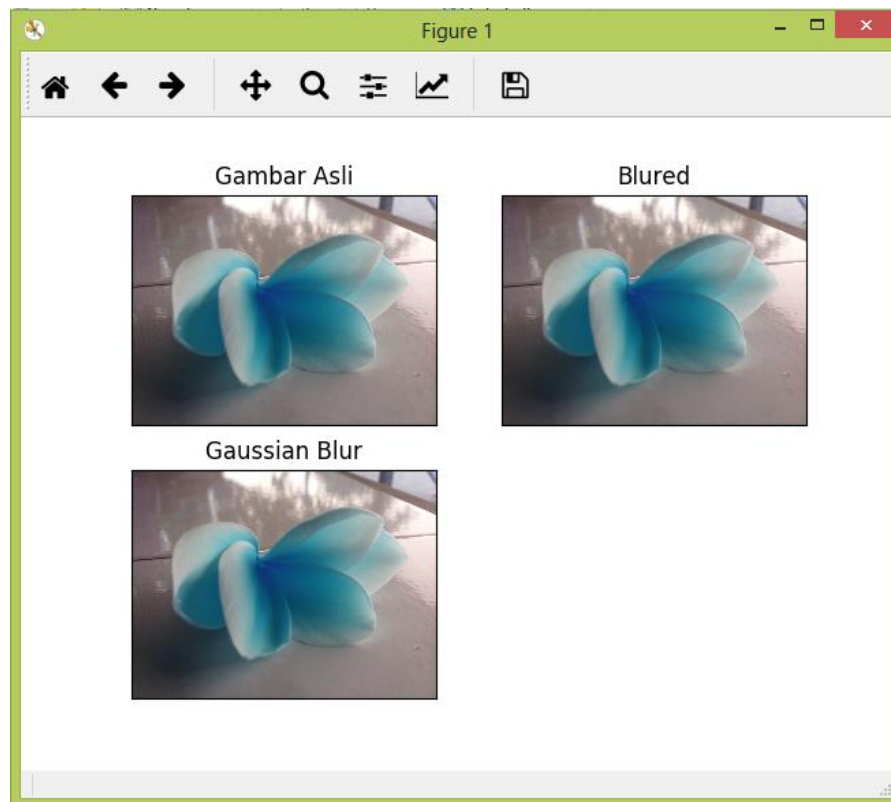
```
12 blur2 = cv2.GaussianBlur(img,(5,5),0)
13 #memberikan efek gaussian blur
```

Untuk memberikan efek Gaussian blurring. setiap titik dari gambar akan terhitung tidak-nol, sehingga seluruh gambar harus dihitung setiap pixelnya. (5,5) adalah ukuran kernel lebar dan tinggi.

```
15 plt.subplot(2,2,1),plt.imshow(img),plt.title('Gambar Asli')
16 plt.xticks([], plt.yticks([]))
17 plt.subplot(2,2,2),plt.imshow(blur),plt.title('Blured')
18 plt.xticks([], plt.yticks([]))
19 plt.subplot(2,2,3),plt.imshow(blur2),plt.title('Gaussian Blur')
20 plt.xticks([], plt.yticks([]))
21 plt.show()
22 #menampilkan gambar dalam satu bingkai
```

Untuk menampilkan gambar ke dalam satu bingkai (figure).

🚦 **Hasil:**



3. Histogram

Histogram digunakan untuk merepresentasikan graphical dari distribusi intensitas gambar. Digunakan pula untuk mengkuantisasi nomor pixel tiap nilai intensitas. Histogram yang saya buat dengan menggunakan OpenCV dengan nama **histogram.py**. Berikut ini source code-nya:

```
1  import cv2
2  import numpy as np
3  from matplotlib import pyplot as plt
4  #untuk mengimport modul atau library
5
6  gray_img = cv2.imread('Pot Bunga.jpg', cv2.IMREAD_GRAYSCALE)
7  cv2.imshow('Pot Bunga', gray_img)
8  #untuk inisialisasi gambar dan mengubah ke skala keabuan
9
10 hist = cv2.calcHist([gray_img],[0],None,[256],[0,256])
11 #fungsi histogram
12
13 plt.hist(gray_img.ravel(),256,[0,256])
14 plt.title('Histogram Skala Keabuan')
15 plt.show()
16 #menampilkan gambar dalam bingkai dengan nama "Histogram Skala Keabuan"
17
18 while True:
19     k = cv2.waitKey(0) & 0xFF
20     if k == 27 : break
21 cv2.destroyAllWindows()
22 #untuk menutup bingkai jika tombol ESC ditekan
```

🔧 Penjelasan:

```
1  import cv2
2  import numpy as np
3  from matplotlib import pyplot as plt
4  #untuk mengimport modul atau library
```

Untuk mengimport modul atau library yang akan digunakan.

```
6  gray_img = cv2.imread('Pot Bunga.jpg', cv2.IMREAD_GRAYSCALE)
7  cv2.imshow('Pot Bunga', gray_img)
8  #untuk inisialisasi gambar dan mengubah ke skala keabuan
9
```

Untuk menginisialisasi gambar lalu mengubahnya kedalam skala keabuan.

```
10 hist = cv2.calcHist([gray_img],[0],None,[256],[0,256])
11 #fungsi histogram
12
```


fungsi untuk histogram. [gray_img], gambar yang digunakan adalah type uint8 atau float32. [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.

```
13 plt.hist(gray_img.ravel(),256,[0,256])
14 plt.title('Histogram Skala Keabuan')
15 plt.show()
16 #menampilkan gambar dalam bingkai dengan nama "Histogram Skala Keabuan"
17
```

Untuk menampilkan gambar ke dalam bingkai (figure) dengan nama “Histogram Skala Keabuan”.

```
18 while True:
19     k = cv2.waitKey(0) & 0xFF
20     if k == 27 : break
21     cv2.destroyAllWindows()
22 #untuk menutup bingkai jika tombol ESC ditekan
```

Perintah untuk menutup bingkai (figure) dengan menekan tombol ESC.

✚ **Hasil:**

