

LAPORAN MATA KULIAH PENGOLAHAN CITRA DIGITAL (PCD)

Dosen Pengampu : Shinta Dwi Anggreni, S. Si.,M.Kom

*“Implementasi Difference Image, Image Averaging, dan histogram equalization
menggunakan VSC”*



Disusun Oleh :

Widya ayunindya poge

F55121071

PROGRAM STUDI S1 TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

FAKULTAS TEKNIK

UNIVERSITAS TADULAKO

2023

A. Difference Image

1. Import package yang digunakan.

```
difference image.py X
difference image.py > ...
1  from PIL import Image, ImageChops
2
```

2. Kemudian menambahkan gambar yang akan diproses.

```
3  img1 = Image.open('gambar.jpg')
4  img2 = Image.open('gambar1.jpg')
```



3. Kemudian memasukan fungsi difference untuk menghitung pertbedaan piksel antara gambar sumber di gambar target, dan fungsi difference memiliki 2 parameter yaitu gambar1 dan gambar2. Fungsi itu akan mengembalikan atau mengeluarkan output gambar.

```
6  diff = ImageChops.difference(img1, img2)
7  if diff.getbbox():
8      diff.show()
```

4. Output

Output yang ditampilkan ada sebagai berikut dimana perbedaan dari 2 gambar ditandai dengan hasil gambar yang bagian berbedanya diberi warna terang agar user mengetahui/menemukan perbedaan dari 2 gambar.



B. Image Averaging

1. Import package yang akan digunakan.

```
Average.py > ...  
1  import cv2  
2  import numpy as np  
3  from skimage.util import random_noise  
4  from matplotlib import pyplot as plt  
5
```

2. Kemudian input gambar yang akan digunakan.

```
6  # Load image yang akan digunakan  
7  img = cv2.imread('gambar.webp')  
8  ori_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)  
9
```

3. Menambahkan gaussian noise ke citra aslinya, yang mana fungsi ini akan menghasilkan citra dengan nilai float.

```
11  noise_img = random_noise(ori_img, mode='gaussian')  
12
```

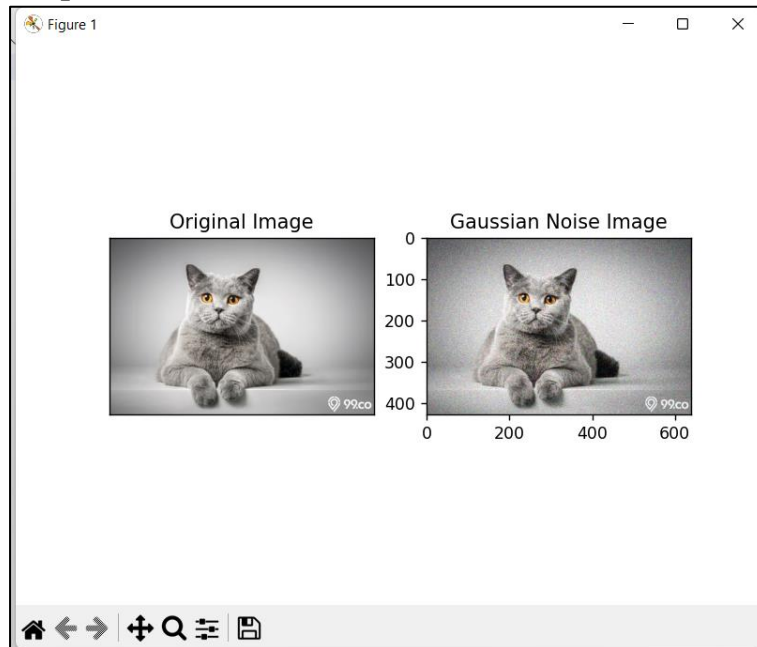
4. Hasil tadi berada pada rentang nilai [0, 1], sehingga perlu diubah menjadi format uint8 dengan rentang nilai [0,255].

```
11  noise_img = random_noise(ori_img, mode='gaussian')  
12
```

5. Terakhir tampilkan citra dengan noise.

```
20  plt.subplot(121), plt.imshow(ori_img), plt.title('Original Image')  
21  plt.xticks([], plt.yticks([]))  
22  plt.subplot(122), plt.imshow(noise_img), plt.title('Gaussian Noise Image')  
23  plt.show()
```

6. Output



C. Histogram Equalization

1. Import package yang akan digunakan.

```
Equalization.py > ...  
1 import cv2  
2 import numpy as np  
3 from matplotlib import pyplot as plt  
4
```

2. Menambahkan gambar yang akan digunakan.

```
6 img = cv2.imread('gambar.webp', 0)  
7
```

3. Menghitung histogram gambar asli.

```
9 hist, bins = np.histogram(img.flatten(), 256, [0, 256])  
10
```

4. Melakukan equalization histogram.

```
12 equ_img = cv2.equalizeHist(img)  
13
```

5. Menghitung histogram gambar yang telah di ekualisasi.

```
15 equ_hist, equ_bins = np.histogram(equ_img.flatten(), 256, [0, 256])  
16
```

6. Menampilkan gambar asli dan gambar yang telah di eksekusi.

```
18 cv2.imshow('Original Image', img)  
19 cv2.imshow('Equalized Image', equ_img)  
20
```

7. Menampilkan grafik.

```
22 plt.figure(figsize=(8, 6))
23 plt.hist(img.flatten(), 256, [0, 256], color='r')
24 plt.hist(equ_img.flatten(), 256, [0, 256], color='b')
25 plt.xlim([0, 256])
26 plt.legend(('Original Image', 'Equalized Image'), loc='upper left')
27 plt.show()
28
29 cv2.waitKey(0)
30 cv2.destroyAllWindows()
```

8. Output

