

STUDI KASUS EDGE DETECTION

Dosen Pengampu: NURSENO BAYU AJI, S. Kom, M. Kom.



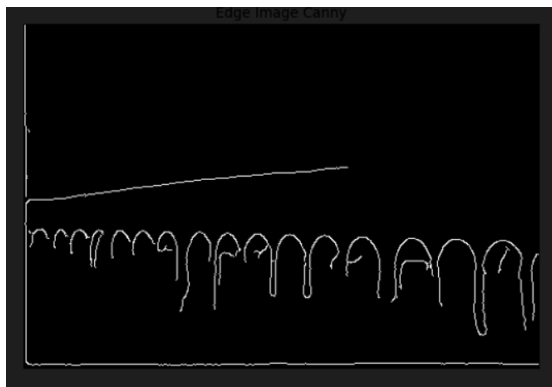
Disusun Oleh

1. Ginanjar Riski Okta Afriansa (4.33.20.0.12)
2. Marcellinus (4.33.20.0.14)
3. Muhammad Athallah Annaafi (4.33.20.0.16)
4. Nandi Sapta Permana (4.33.20.0.20)
5. Rizki Adi Hidayat (4.33.20.0.22)

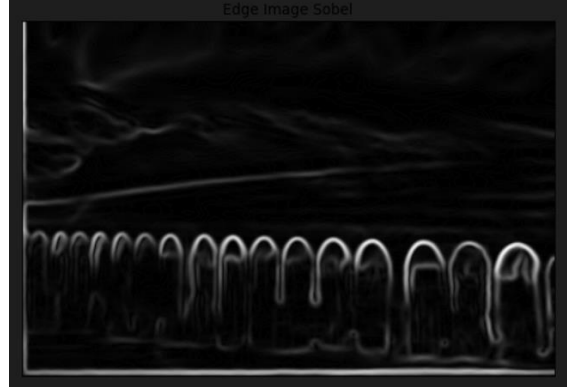
TI-2A

**PROGRAM STUDI TEKNOLOGI REKAYASA KOMPUTER
JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI SEMARANG
2022**

No.	Hasil
1.	<p>Coding :</p> <pre> import numpy as np import cv2 as cv from matplotlib import pyplot as plt from skimage.filters import sobel, prewitt, roberts img = cv.imread('6 change.png',0) new_image = cv.fastNlMeansDenoising(img, None, 20,7,20) edges = cv.Canny(new_image,20,95) edges2 = sobel(new_image) plt.figure(num=None, figsize=(8, 6), dpi=80),plt.imshow(img,cmap = 'gray') plt.title('Original Image'), plt.xticks([]), plt.yticks([]) plt.figure(num=None, figsize=(8, 6), dpi=80),plt.imshow(new_image,cmap = 'gray') plt.title('Noiseless'), plt.xticks([]), plt.yticks([]) plt.figure(num=None, figsize=(8, 6), dpi=80),plt.imshow(edges,cmap = 'gray') plt.title('Edge Image Canny'), plt.xticks([]), plt.yticks([]) plt.figure(num=None, figsize=(8, 6), dpi=80),plt.imshow(edges2,cmap = 'gray') plt.title('Edge Image Sobel'), plt.xticks([]), plt.yticks([]) plt.show() </pre> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;"> <p>Original Image</p>  </div> <div style="text-align: center;"> <p>Noiseles</p>  </div> </div>



Edge image Canny



Edge image Sobel

```
from skimage.metrics import structural_similarity as ssim
import matplotlib.pyplot as plt
import numpy as np
import cv2

def mse(imageA, imageB):
    # the 'Mean Squared Error' between the two images
    err = np.sum((imageA.astype("float") - imageB.astype("float")) ** 2)
    err /= float(imageA.shape[0] * imageA.shape[1])

    # return the MSE, the lower the error
    return err

def compare_images(imageA, imageB, title):
    # compute the mean squared error
    m = mse(imageA, imageB)
    # setup the figure
    fig = plt.figure(title)
    plt.title(title)
    plt.suptitle("MSE: %.2f" % (m))
    # show first image
    ax = fig.add_subplot(1, 2, 1)
    plt.imshow(imageA, cmap = plt.cm.gray)
    plt.axis("off")
    # show the second image
    ax = fig.add_subplot(1, 2, 2)
    plt.imshow(imageB, cmap = plt.cm.gray)
    plt.axis("off")
    # show the images
    plt.show()

# hasil canny
```

```

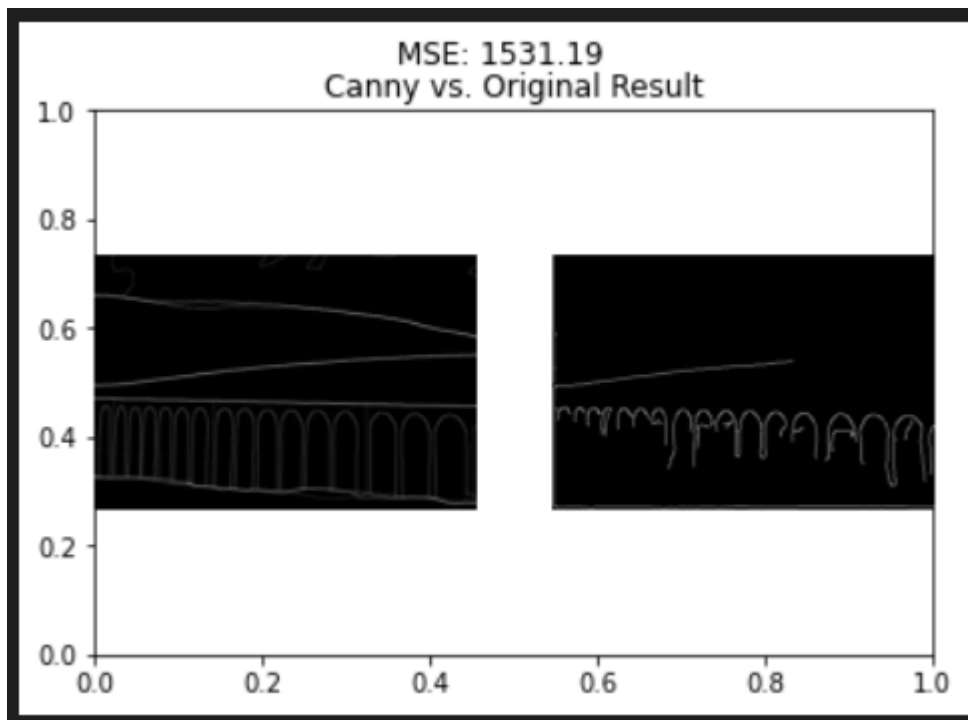
canny = edges

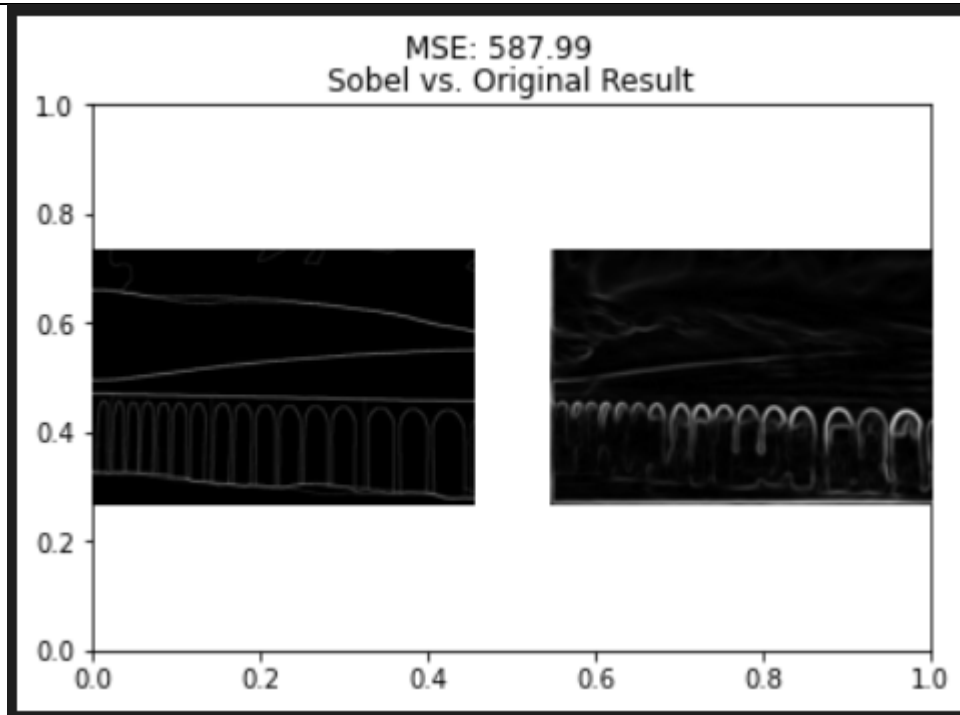
# hasil sobel
sobel = edges2

# hasil asli
result = cv.imread('6 result.jpg',0)

# compare images
compare_images(result, canny, "Canny vs. Original Result")
compare_images(result, sobel, "Sobel vs. Original Result")

```





Penjelasan :

Dari percobaan yang kami lakukan dapat dilihat hasil yang hampir mirip dari gambar yang ingin kita capai adalah dengan metode Sobel, karena memiliki jumlah MSE yang lebih kecil. Pada perbandingan diatas hasil dari Canny memang lebih jelas untuk garisnya akan tetapi masih banyak benda yang seharusnya muncul tetapi tidak muncul, sedangkan dengan metode Sobel kami berhasil memunculkan beberapa garis yang kami tuju walaupun kurang jelas atau solid.