

```
from google.colab import files
uploaded = files.upload()
```



Pilih File produk.jpg

- **produk.jpg**(image/jpeg) - 38346 bytes, last modified: 27/6/2025 - 100% done  
Saving produk.jpg to produk.jpg

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Baca gambar
img = cv2.imread('produk.jpg')
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

```
# Fungsi buat nampilin gambar
def show_img(imgs, titles, rows=1, cols=2):
    plt.figure(figsize=(12, 6))
    for i in range(len(imgs)):
        plt.subplot(rows, cols, i+1)
        plt.imshow(imgs[i])
        plt.title(titles[i])
        plt.axis('off')
    plt.tight_layout()
    plt.show()
```

```
# Tambah brightness
bright = cv2.convertScaleAbs(img_rgb, alpha=1, beta=60) # beta = terang
```

```
show_img([img_rgb, bright], ['Original', 'Brightness +60'])
```



Original



Brightness +60



```
# Tambah kontras
contrast = cv2.convertScaleAbs(img_rgb, alpha=1.8, beta=0) # alpha = kontras
```

```
show_img([img_rgb, contrast], ['Original', 'Contrast x1.8'])
```

◆ Apa yang bisa saya bantu buat untuk Anda?





Original



Contrast x1.8



```
# Filter sharpening
kernel = np.array([[0, -1, 0],
                  [-1, 5, -1],
                  [0, -1, 0]])
sharpen = cv2.filter2D(img_rgb, -1, kernel)

show_img([img_rgb, sharpen], ['Original', 'Sharpened'])
```



Original



Sharpened



```
blur = cv2.GaussianBlur(img_rgb, (11, 11), 0)

show_img([img_rgb, blur], ['Original', 'Gaussian Blur'])
```



Original



Gaussian Blur



```
edges = cv2.Canny(img, 100, 200)
```

```
plt.figure(figsize=(6,6))  
plt.imshow(edges, cmap='gray')  
plt.title('Canny Edge Detection')  
plt.axis('off')  
plt.show()
```



Canny Edge Detection

