MODUL 6

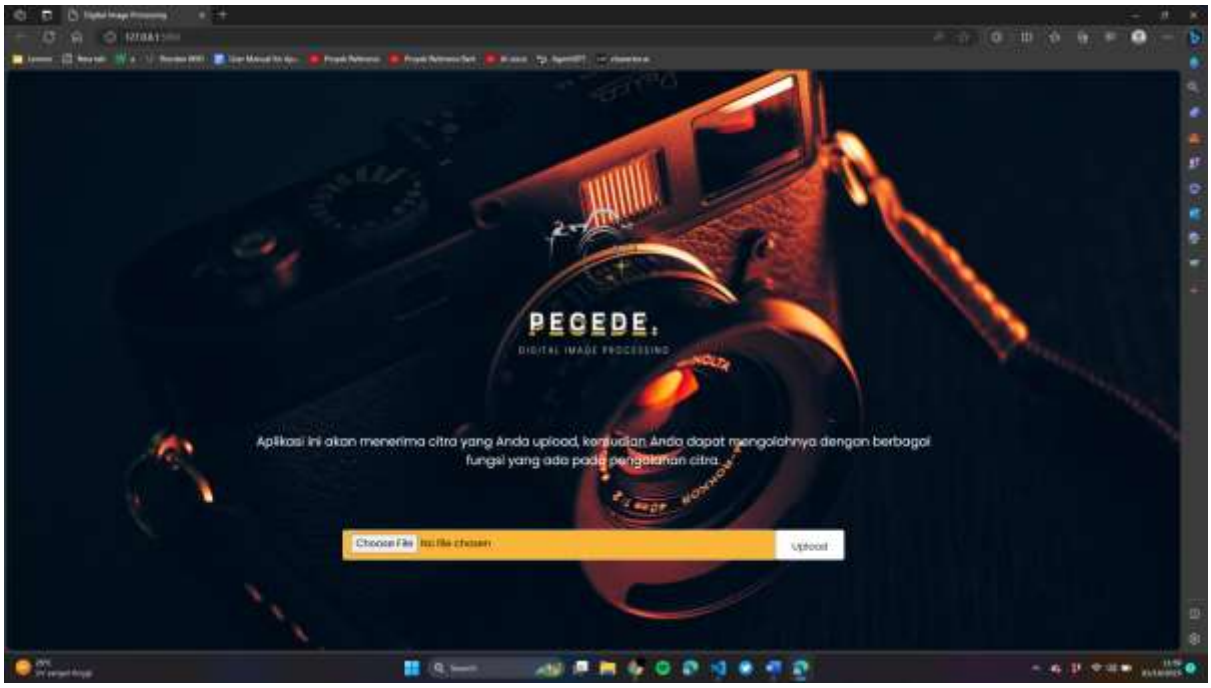
PENGOLAHAN CITRA DIGITAL
MANIPULASI HISTOGRAM CITRA

D3/D4 TEKNIK INFORMATIKA
JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA
POLITEKNIK NEGERI BANDUNG

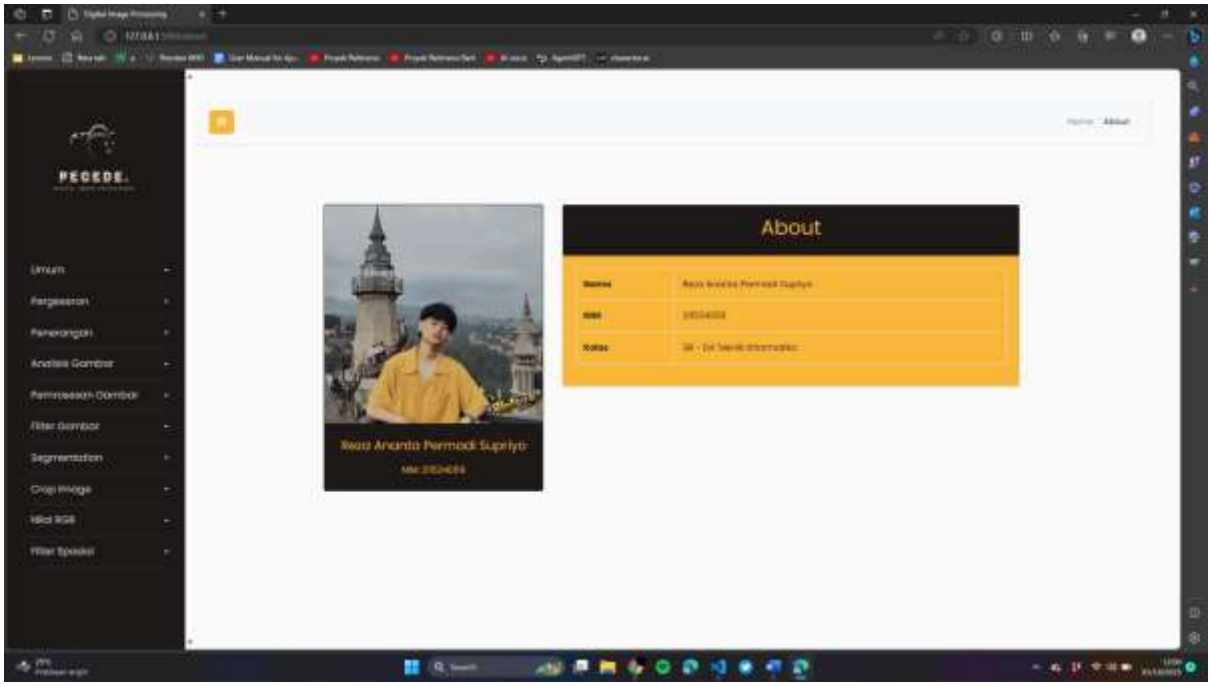
REZA 059 | PENGOLAHAN CITRA DIGITAL | SEPTEMBER, 25 2023

PRAKTIKUM 6

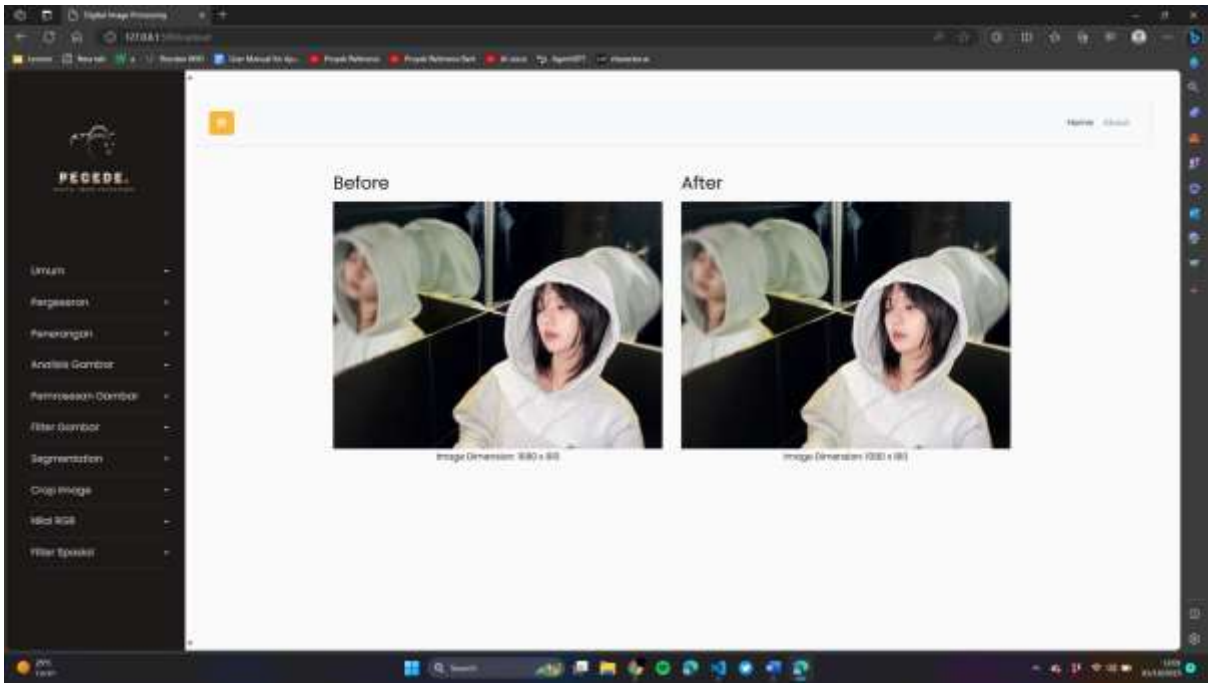
- Upload File



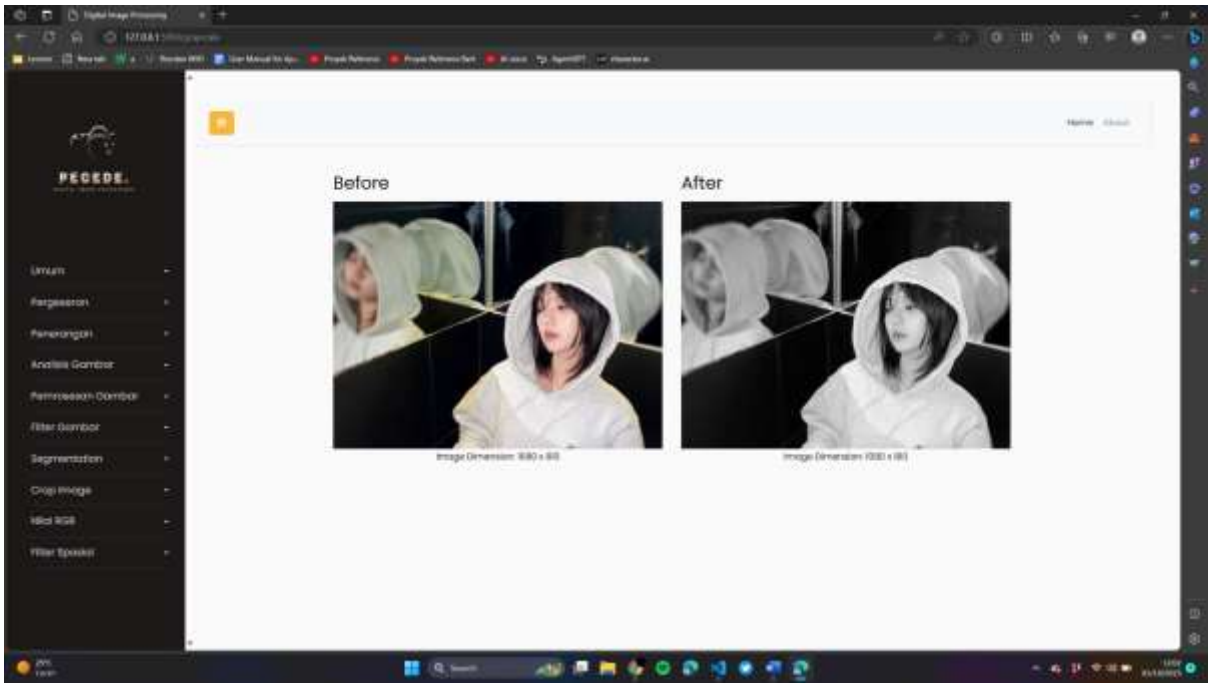
- Halaman About



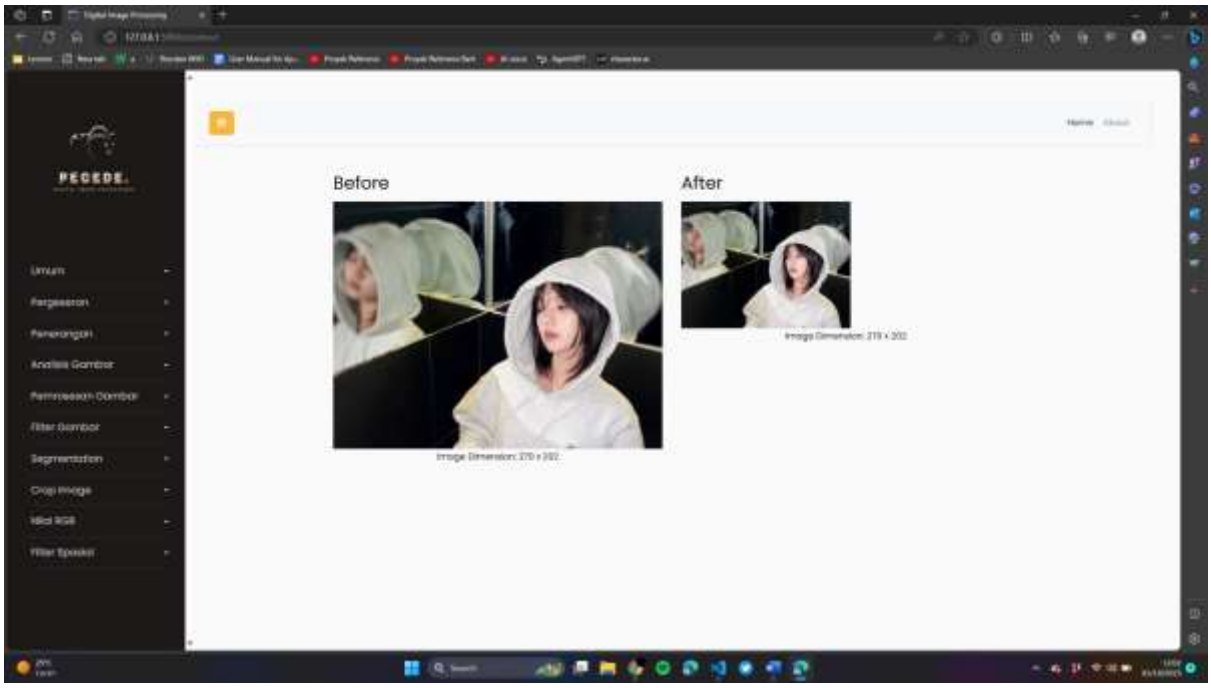
- Tampilan pertama setelah upload gambar (menampilkan gambar normal dengan ukuran dimensinya)



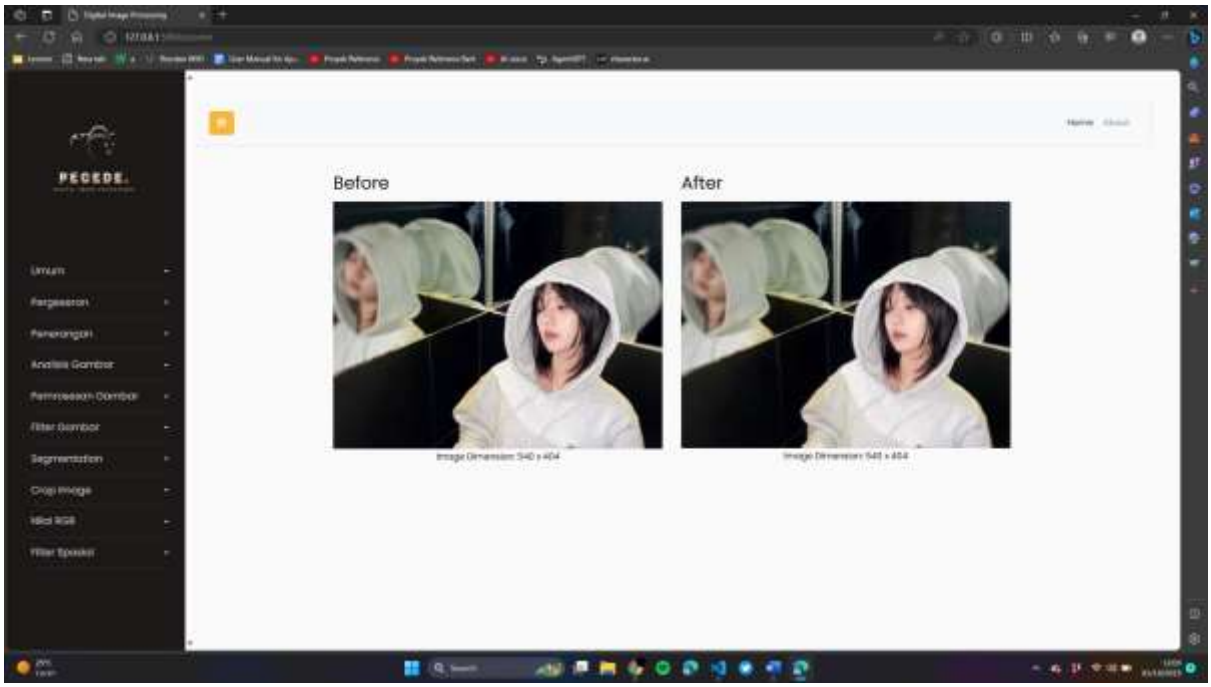
- Grayscale



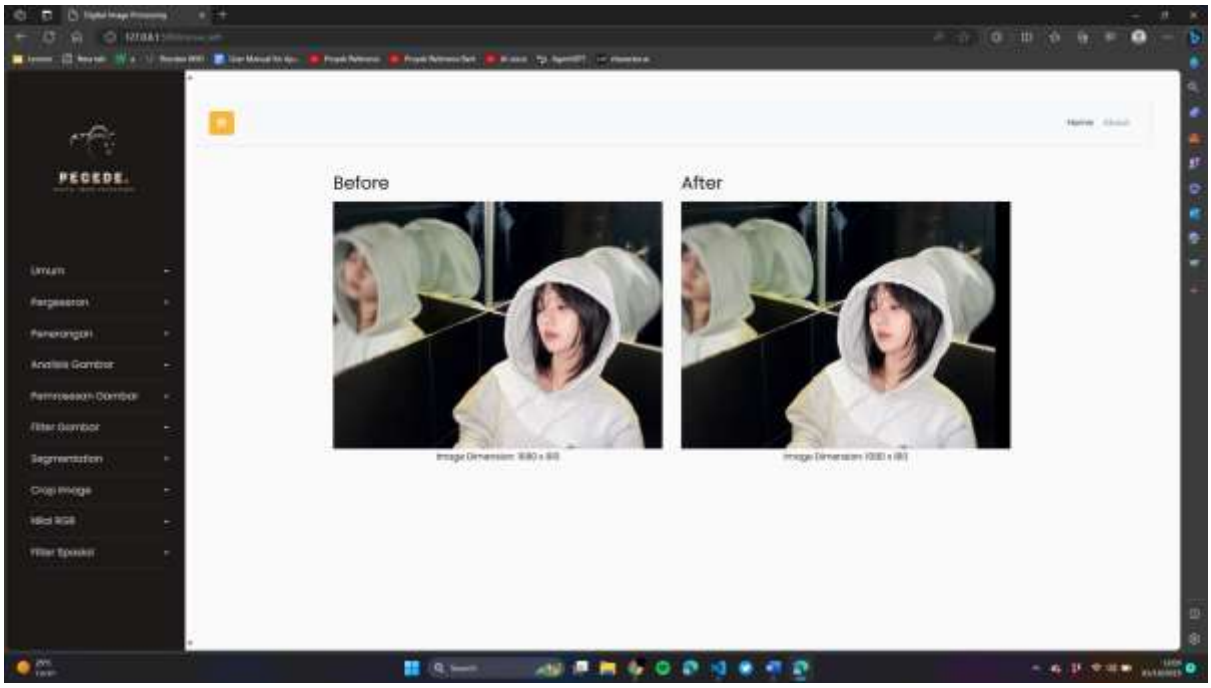
- Zoom Out



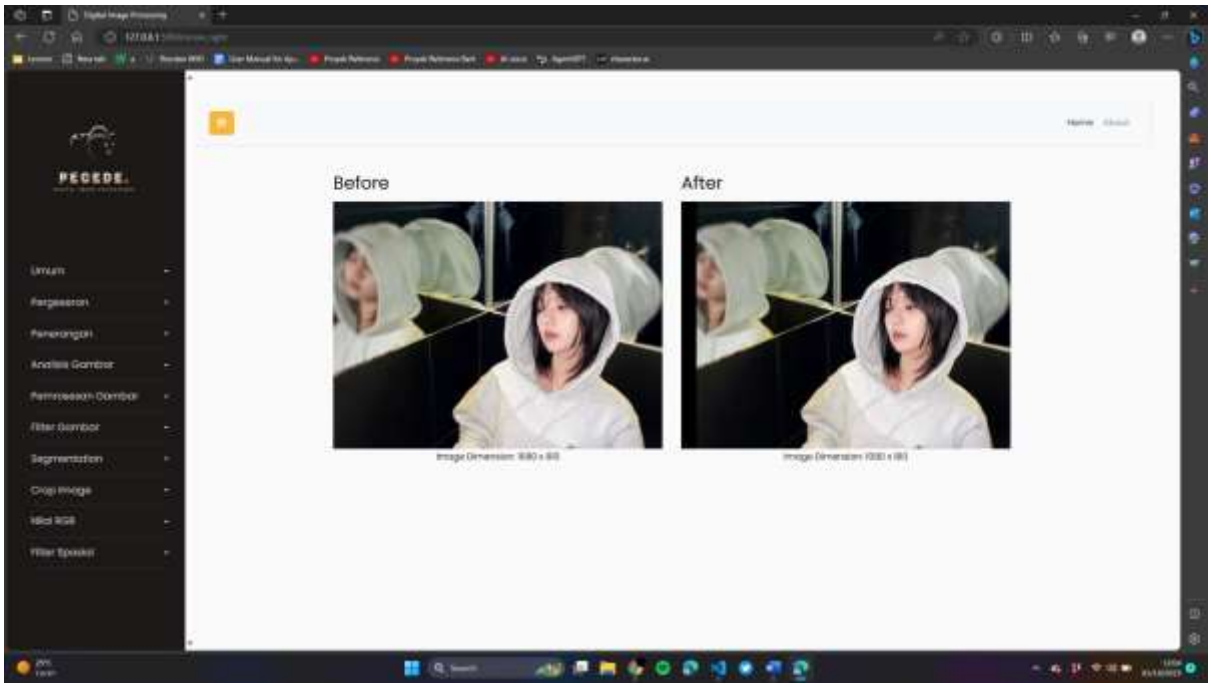
- Zoom In



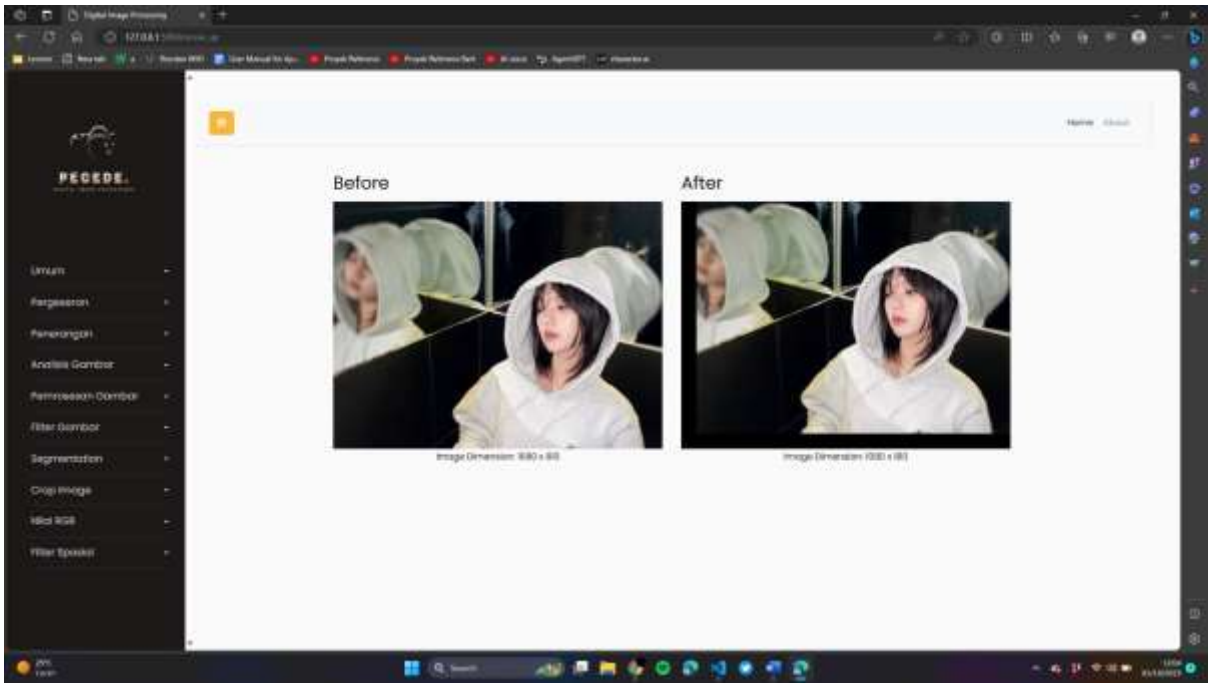
- Geser Kiri



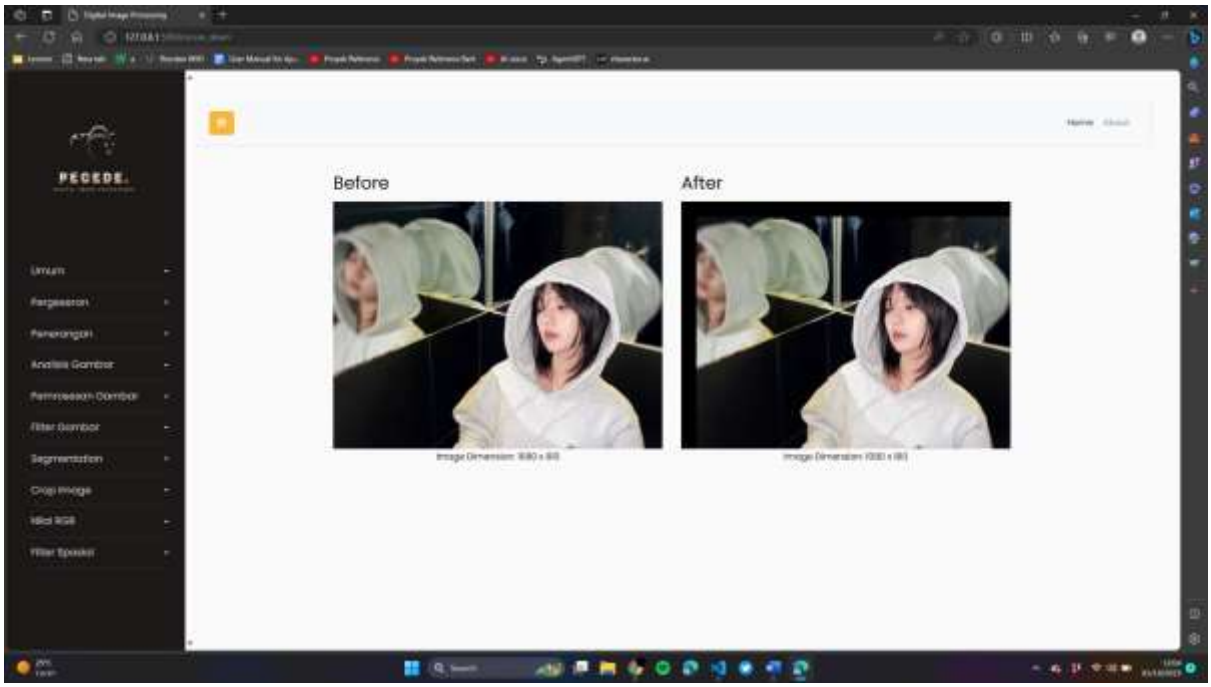
- Geser Kanan



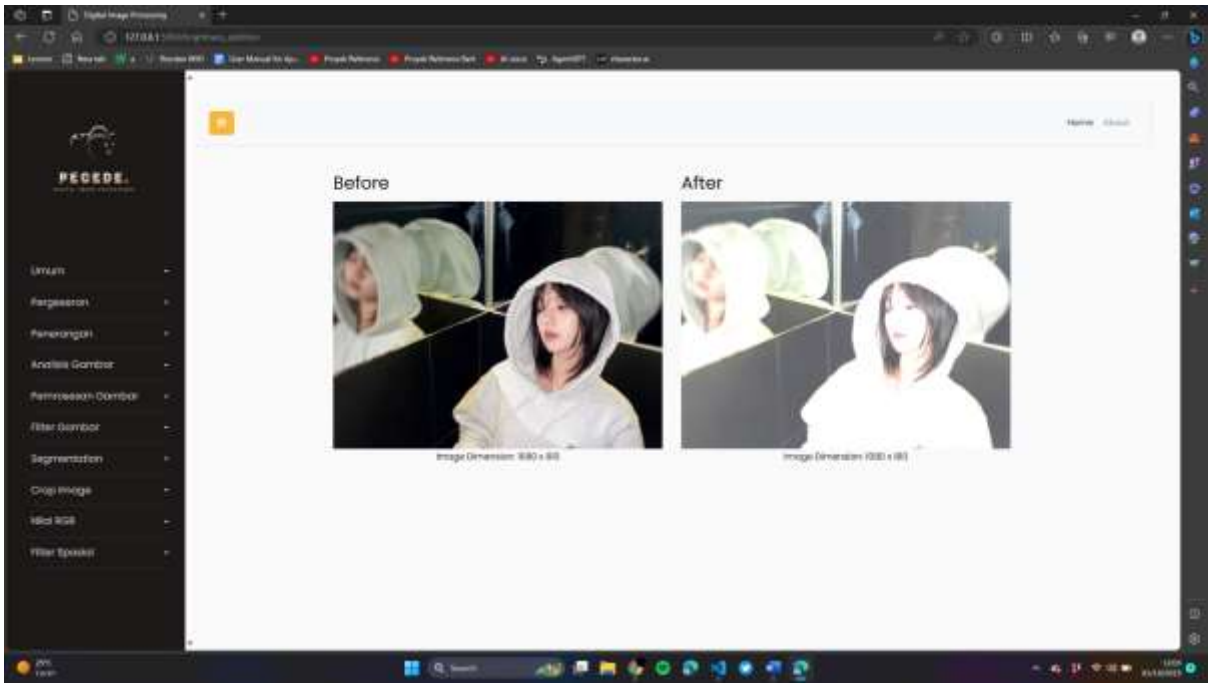
- Geser Atas



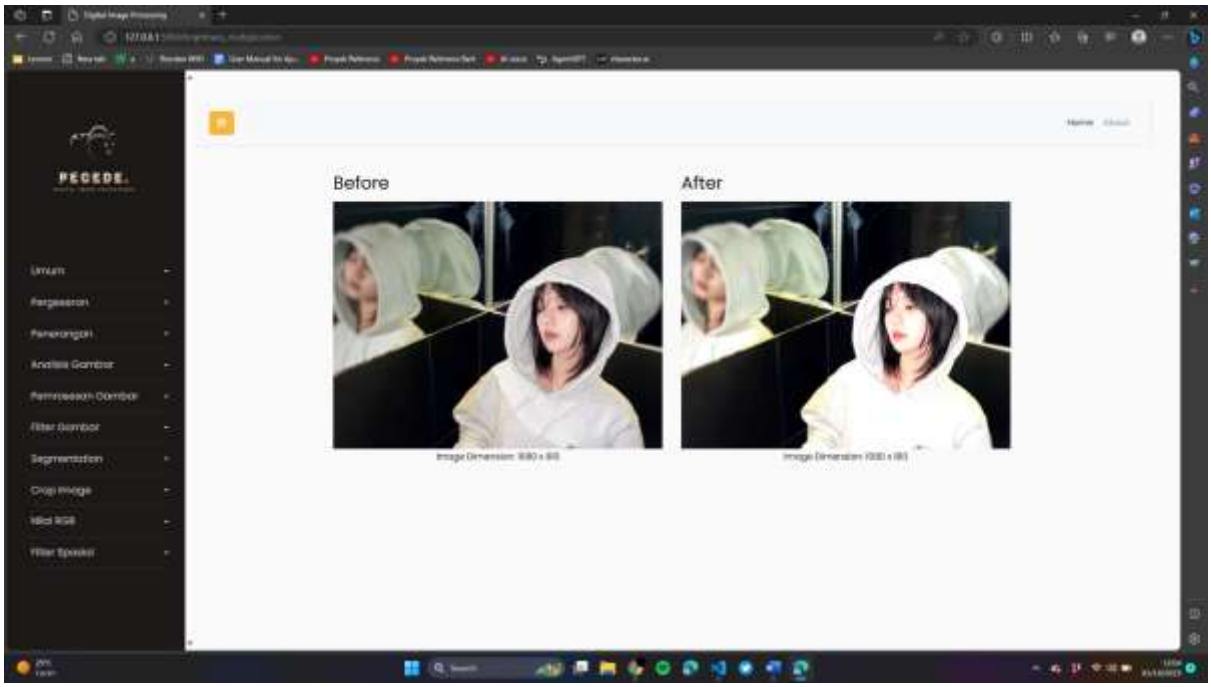
- Geser Bawah



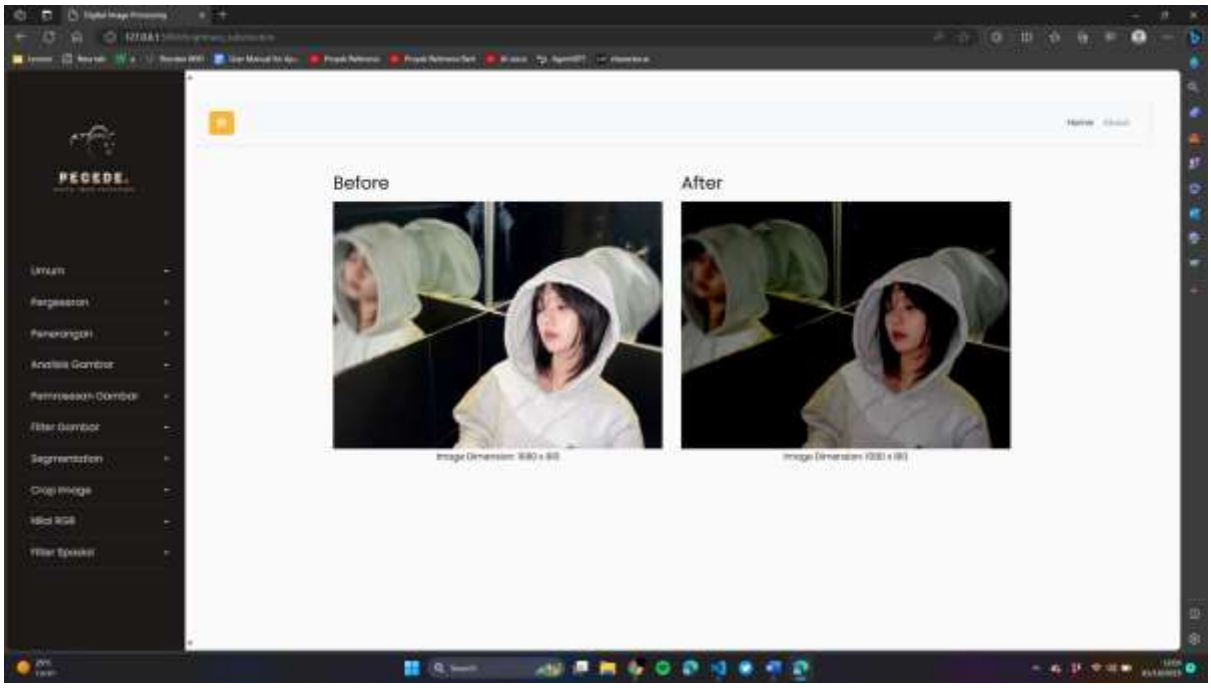
- Terang (+)



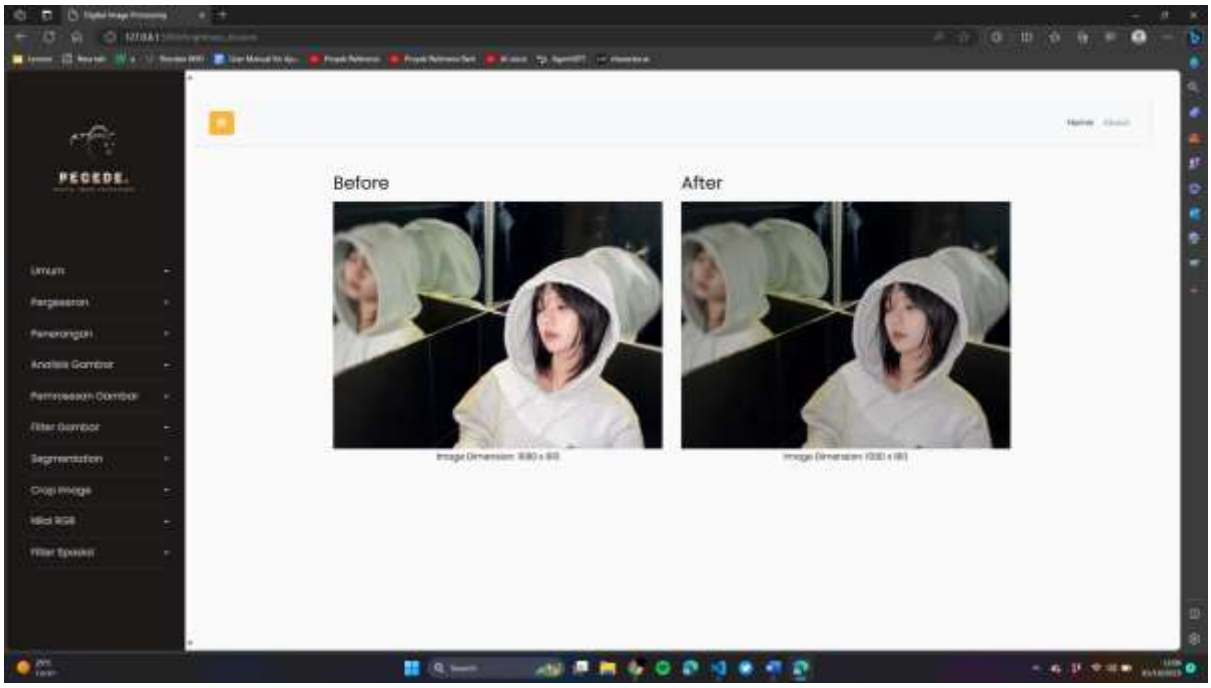
- Terang (*)



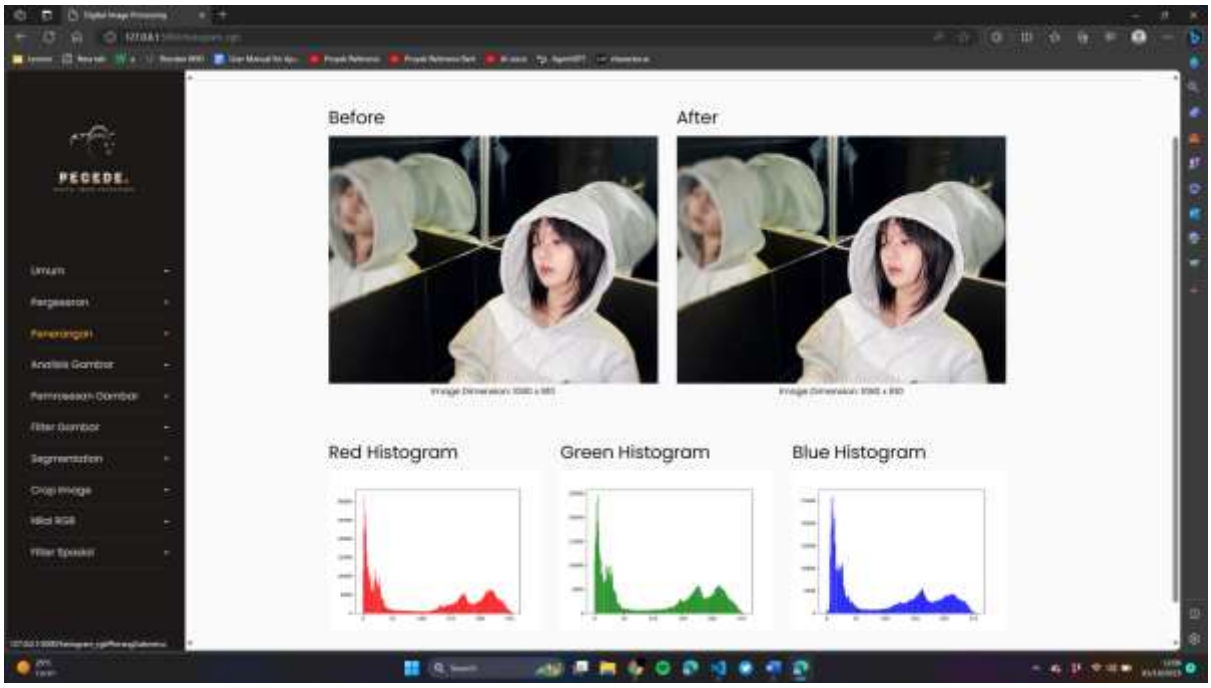
- Gelap (-)



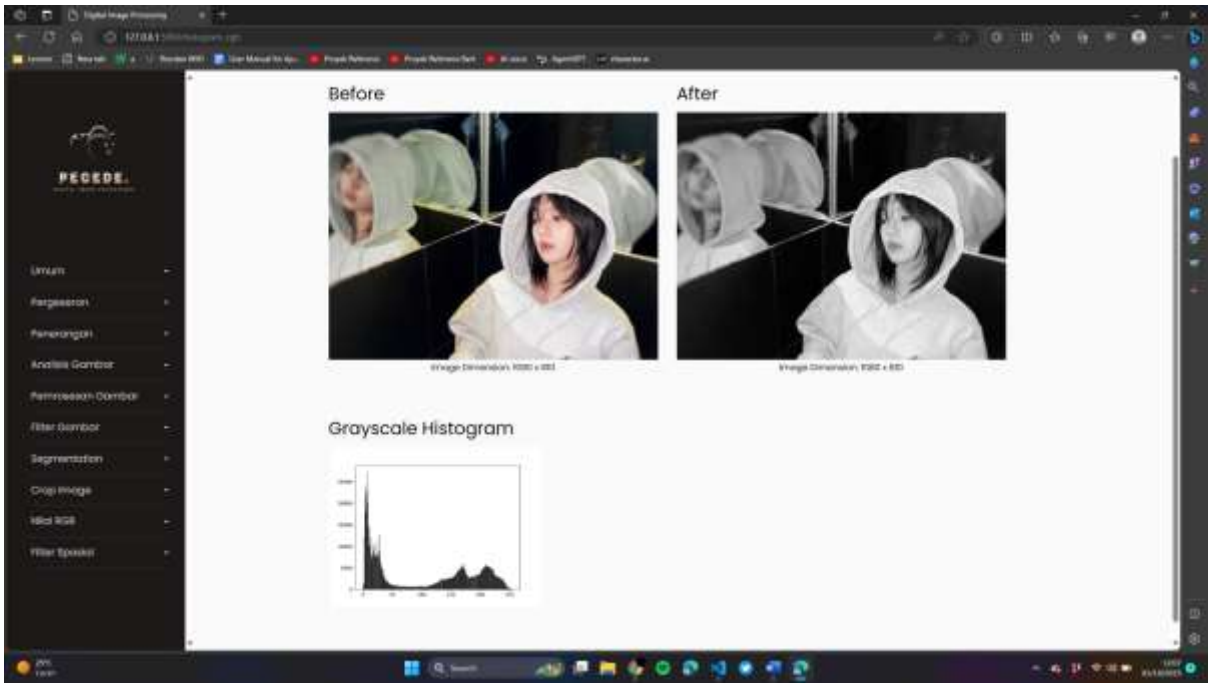
- Gelap (/)



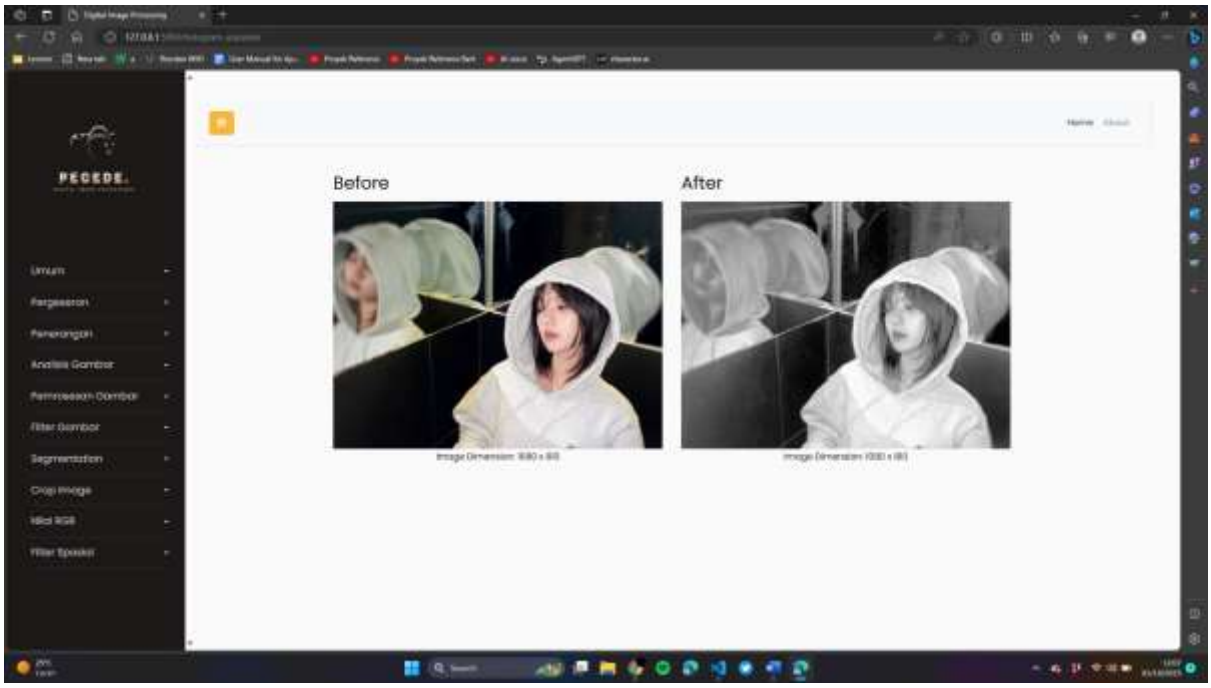
- Histogram RGB



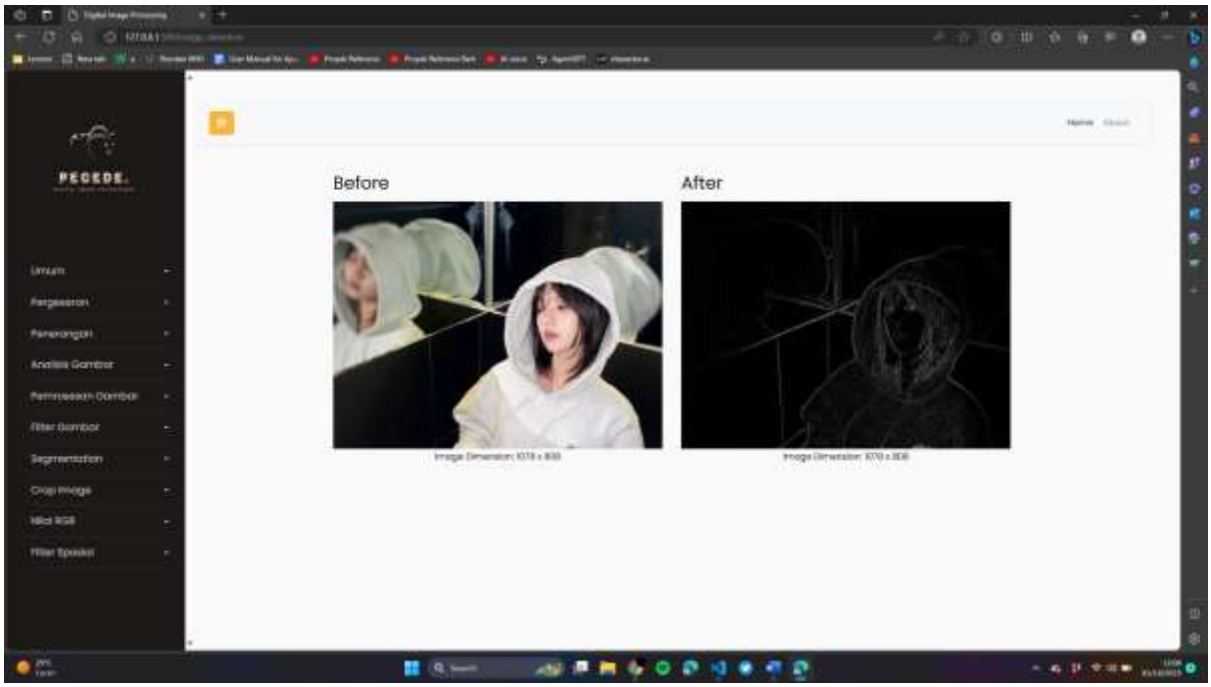
- Histogram Greyscale



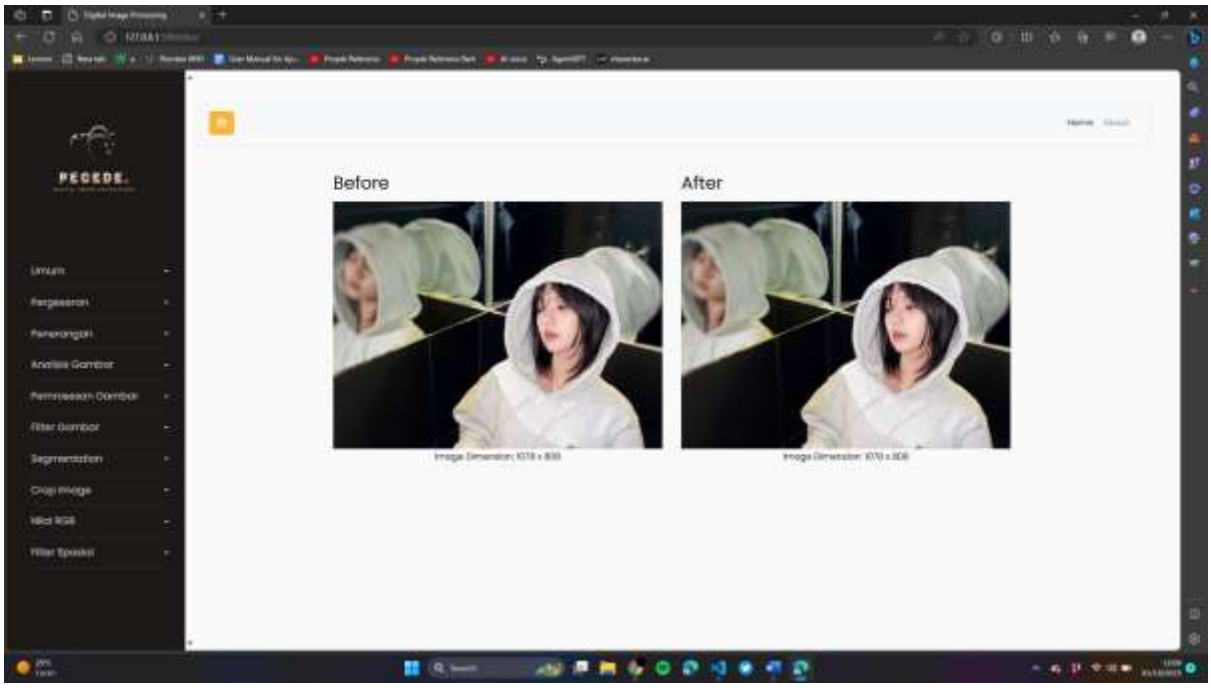
- Histogram Equalizer



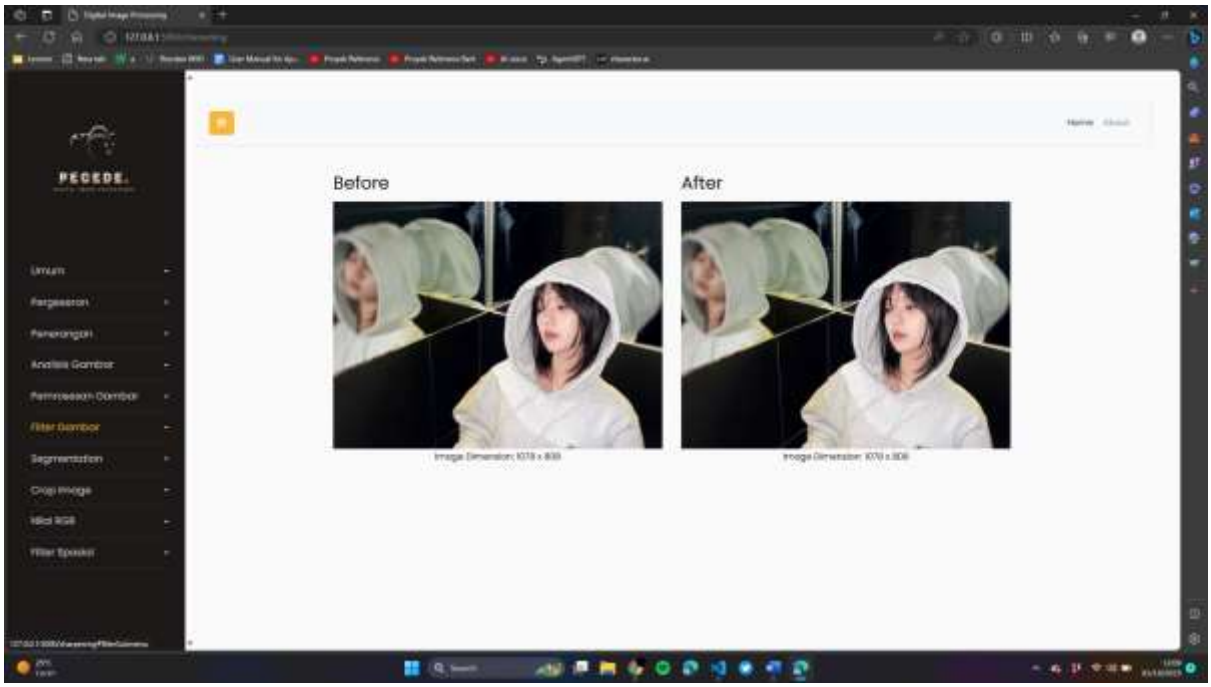
- Edge Detection



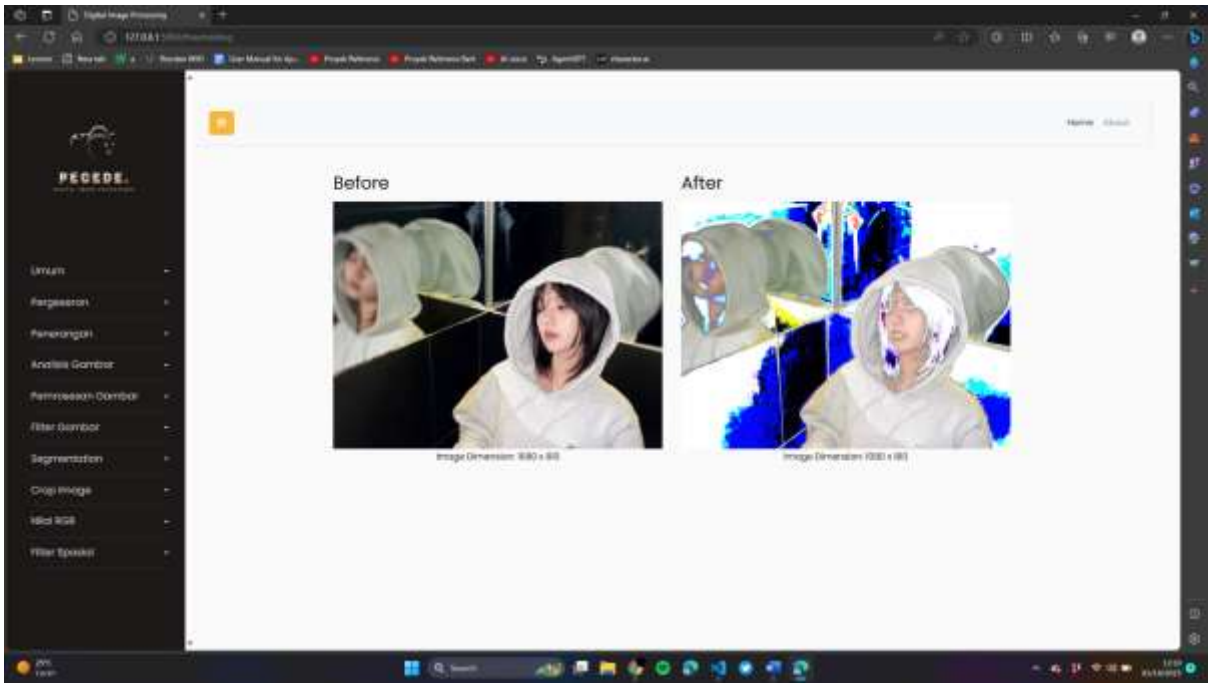
- Blur



- Sharpening



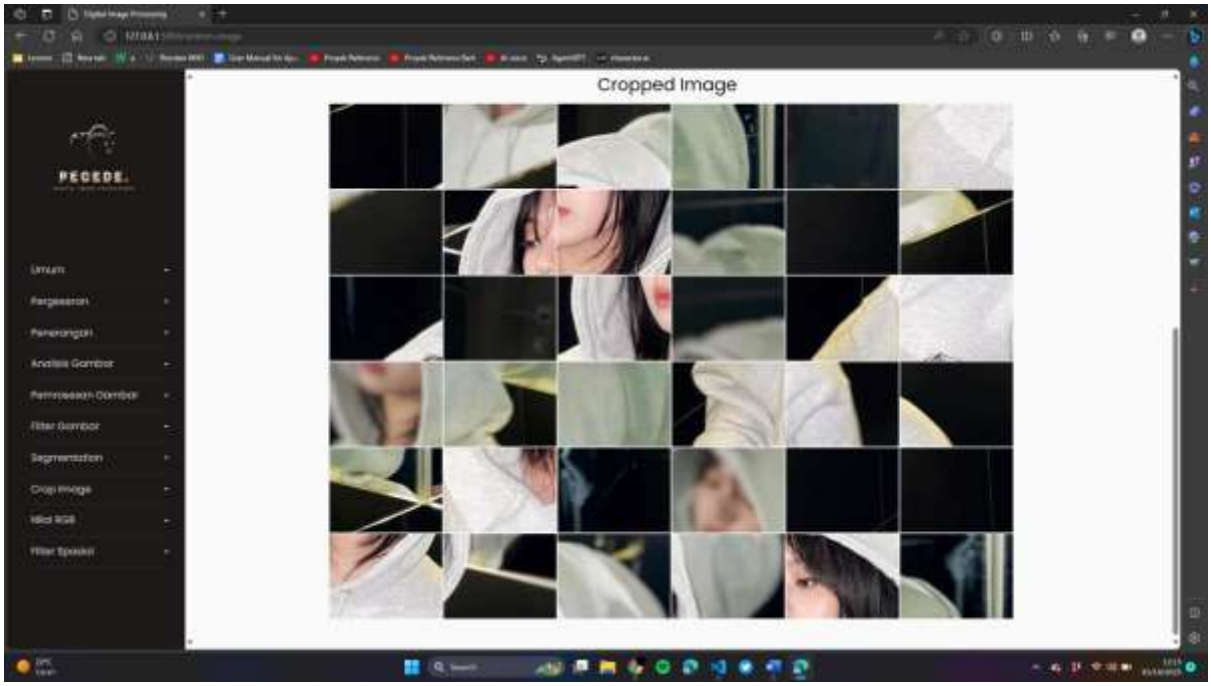
- Thresholding (10 - 100)



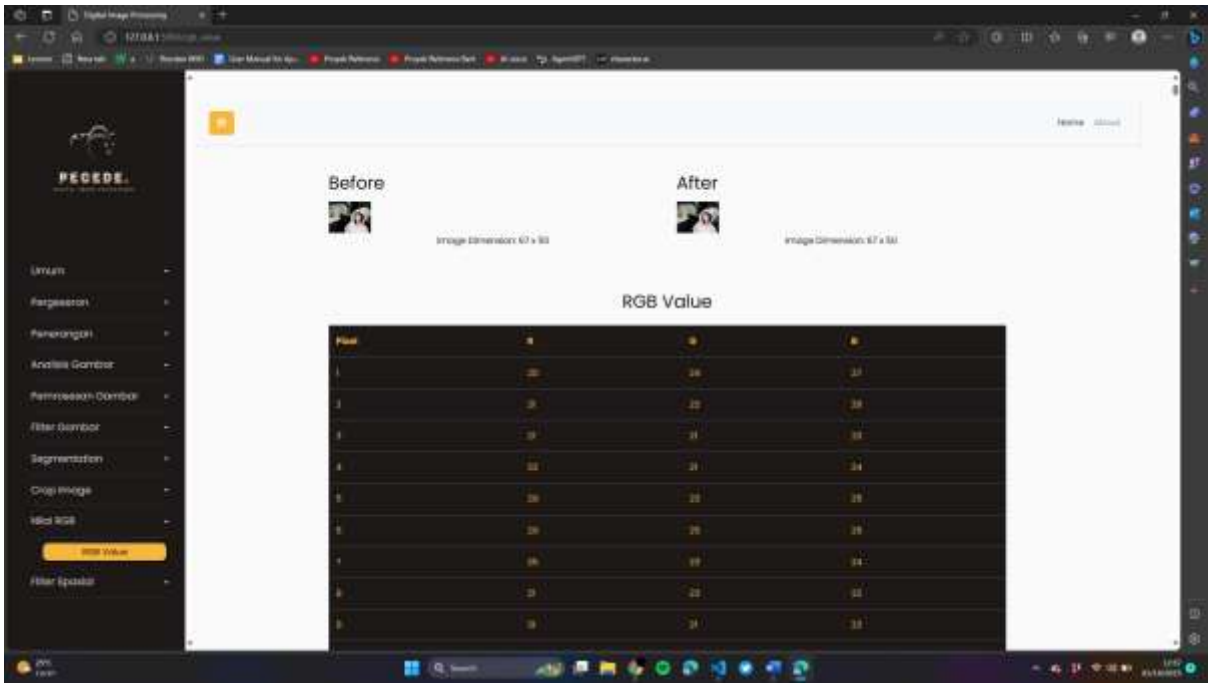
PRAKTIKUM 5 (Challenge)

- Crop Image (7)

- Random Crop (Puzzling = 6)



- Nilai RGB



PRAKTIKUM 6

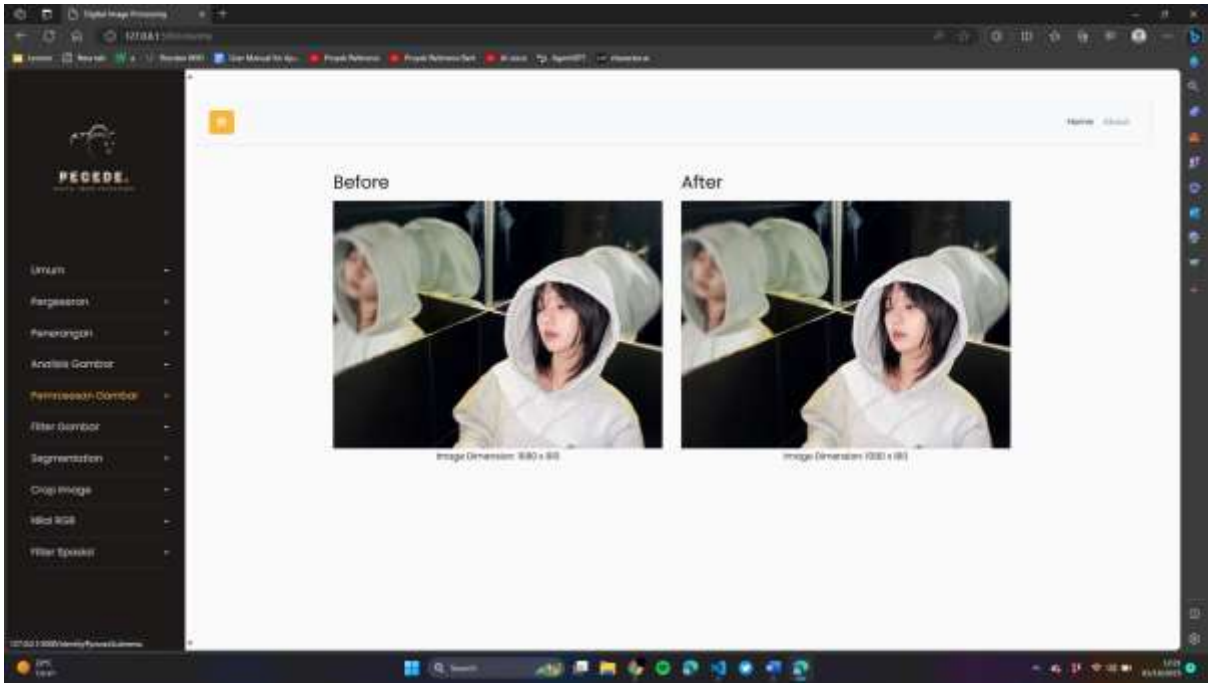
- Identity

```
def identity():
    img = cv2.imread("static/img/img_normal.jpg")

    kernel = np.array([[0, 0, 0],
                       [0, 1, 0],
                       [0, 0, 0]])

    identity = cv2.filter2D(img, -1, kernel)

    cv2.imwrite("static/img/img_now.jpg", identity)
```

Fungsi ini digunakan untuk menerapkan operasi identitas pada gambar yang diberikan. Operasi identitas adalah jenis tindakan yang tidak mengubah tampilan gambar dengan cara apapun. Dalam situasi ini, operasi identitas digunakan sebagai ilustrasi atau contoh, dan tidak akan menghasilkan perubahan visual apapun pada gambar tersebut.

```
kernel = np.array([[0, 0, 0],
                   [0, 1, 0],
                   [0, 0, 0]])
```

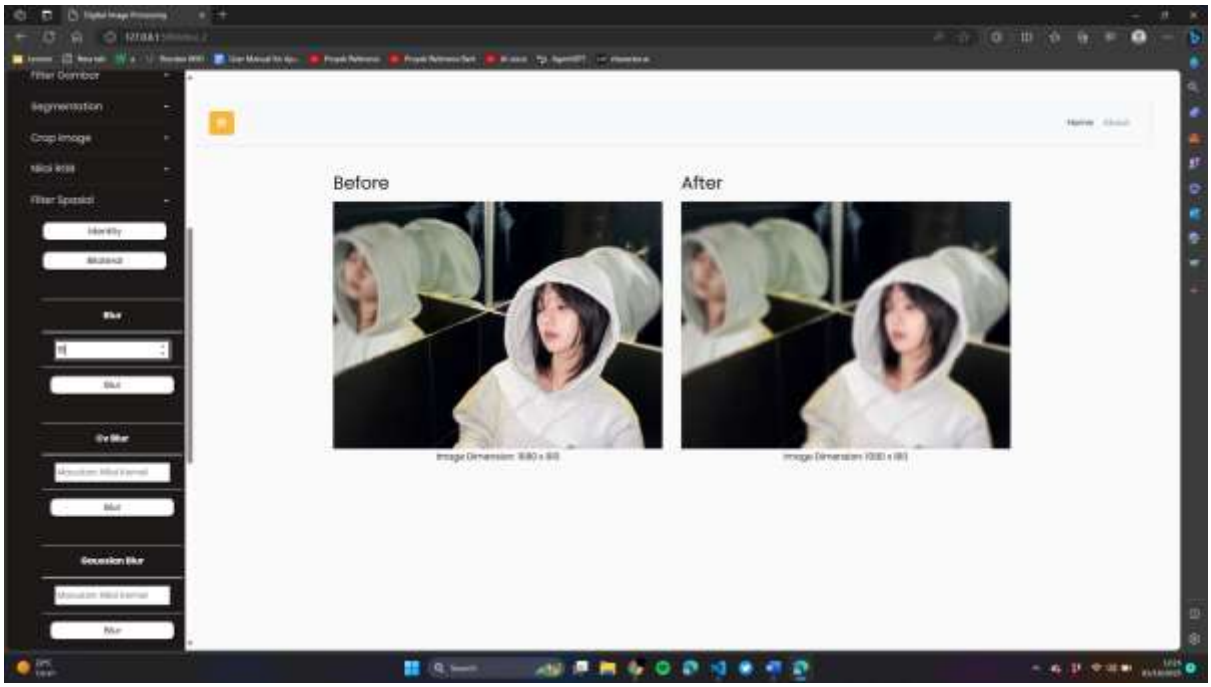
Baris ini menginisialisasi kernel yang akan digunakan dalam operasi konvolusi. Kernel yang digunakan adalah kernel identitas, yang tidak akan mengubah gambar. Kernel ini terdiri dari matriks 3x3 dengan nilai 1 di tengahnya dan nilai 0 di sekitarnya.



- Blur

```
def blur_2(kernel):
    img = cv2.imread("static/img/img_normal.jpg")

    kernel = np.ones((kernel, kernel), np.float32) / (kernel*kernel)

    blur = cv2.filter2D(src=img, ddepth=-1, kernel= kernel)

    cv2.imwrite("static/img/img_now.jpg", blur)
```

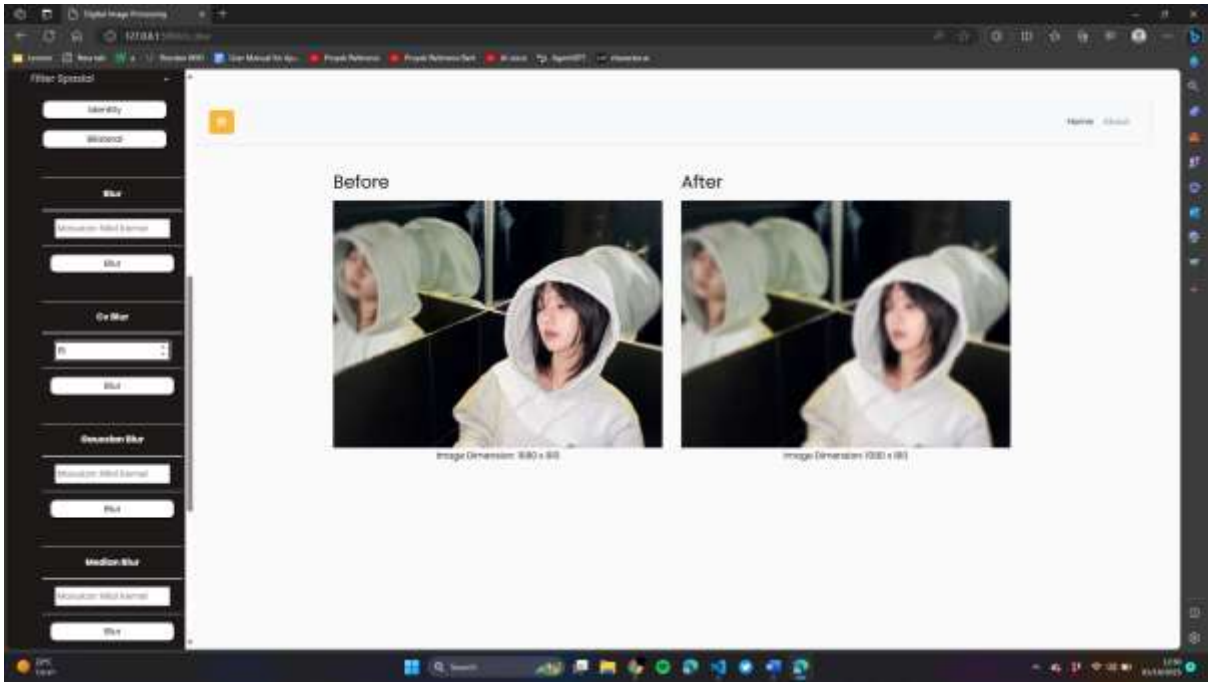
Tujuan dari fungsi ini adalah untuk memberikan efek blur pada gambar yang diberikan. Efek blur ini menyebabkan gambar menjadi lebih kabur atau kurang tajam dengan cara melakukan penghalusan nilai piksel di sekitar setiap piksel dalam gambar. Dalam fungsi blur ini, pengguna memiliki opsi untuk memasukkan kernel melalui halaman webnya yang akan digunakan untuk memproses fungsi tersebut.



- Cv2 Blur

```
def cv_blur(kernel):
    img = cv2.imread("static/img/img_normal.jpg")

    cv_blur = cv2.blur(src=img, ksize=(kernel, kernel))

    cv2.imwrite("static/img/img_now.jpg", cv_blur)
```

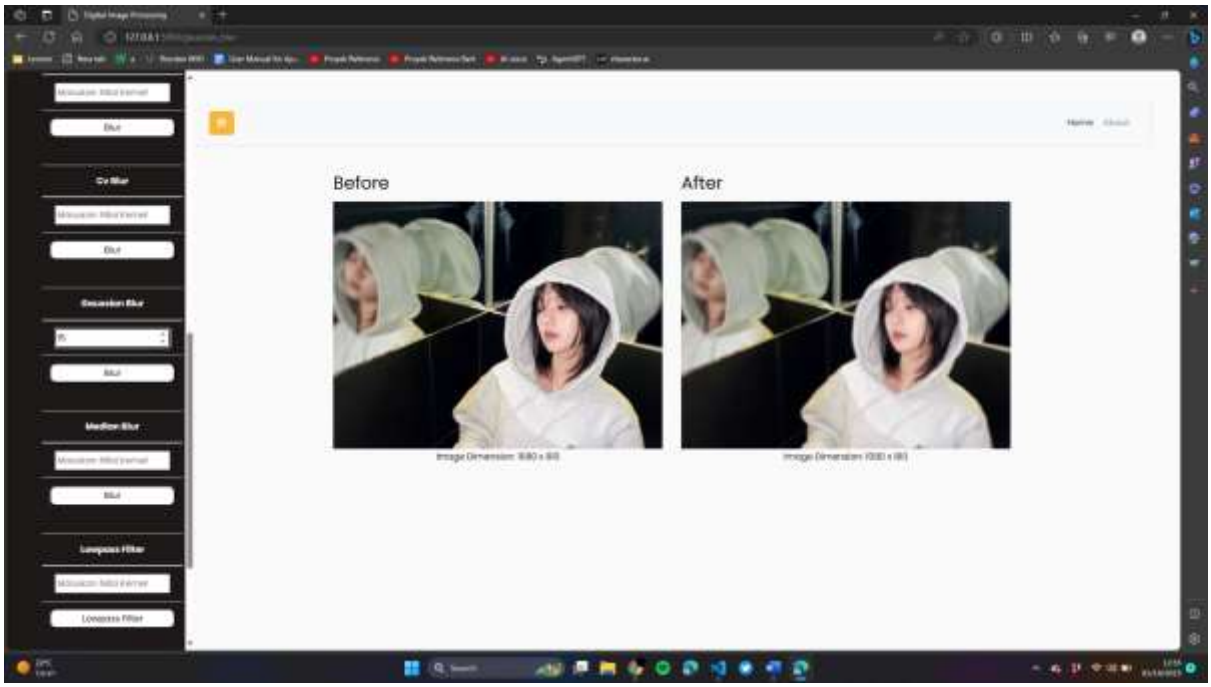
Fungsi di atas nampaknya adalah bagian dari sebuah program yang menggunakan OpenCV (Open Source Computer Vision Library) untuk melakukan proses pengaburan (blur) pada sebuah gambar. Dalam fungsi blur ini, pengguna memiliki opsi untuk memasukkan kernel melalui halaman webnya yang akan digunakan untuk memproses fungsi tersebut.



- Gaussian Blur

```
def gaussian_blur(kernel):
    img = cv2.imread("static/img/img_normal.jpg")

    cv_gaussianblur = cv2.GaussianBlur(src=img,ksize=(kernel,kernel),sigmaX=0)

    cv2.imwrite("static/img/img_now.jpg", cv_gaussianblur)
```

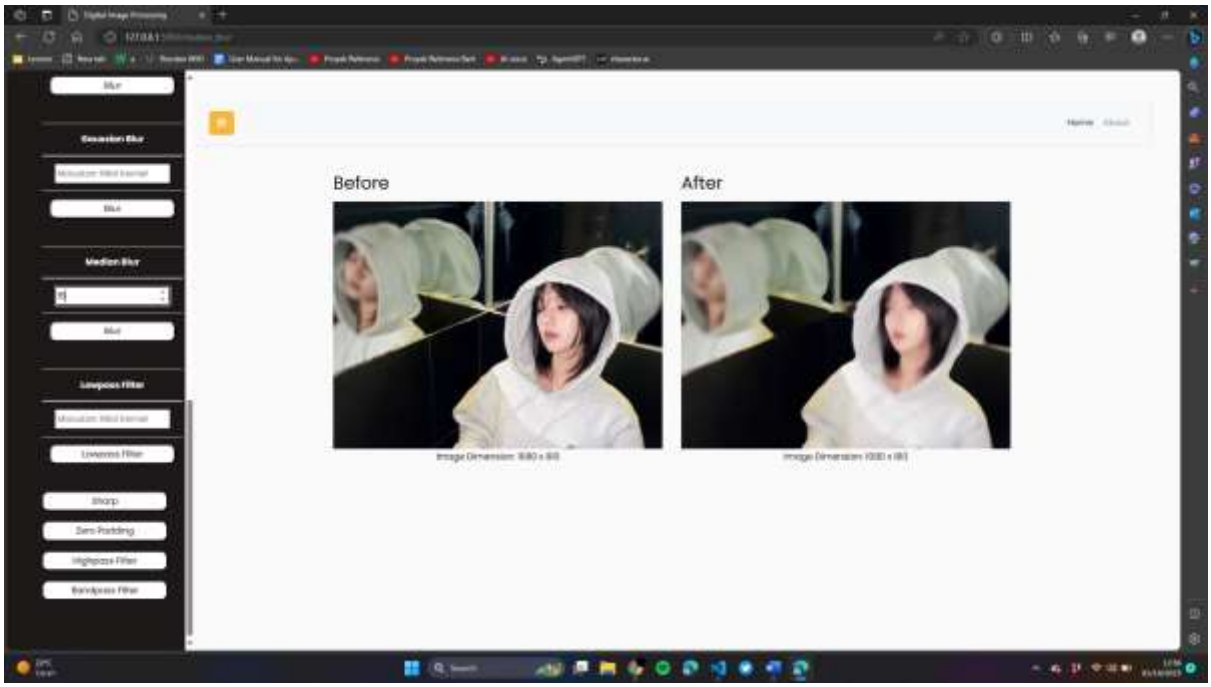
Fungsi ini memiliki tujuan untuk menerapkan efek Gaussian blur pada gambar. Gaussian blur adalah salah satu jenis efek blur yang menciptakan tampilan gambar yang lebih halus dengan merata-ratakan nilai piksel di sekitar setiap piksel berdasarkan distribusi Gaussian. Dalam fungsi Gaussian Blur ini, pengguna memiliki kemampuan untuk memasukkan kernel (dengan syarat harus ganjil) melalui halaman webnya yang akan digunakan untuk memproses fungsi tersebut.



- Median Blur

```
def median_blur(kernel):
    img = cv2.imread("static/img/img_normal.jpg")

    median_blur = cv2.medianBlur(src=img,ksize=kernel)

    cv2.imwrite("static/img/img_now.jpg", median_blur)
```

Fungsi ini memiliki tujuan untuk menerapkan efek median blur pada gambar. Median blur adalah jenis efek blur yang digunakan untuk menghilangkan noise dan detail kecil dari gambar dengan menggantikan nilai piksel dengan nilai median dari piksel-piksel di sekitarnya. Dalam fungsi Median Blur ini, pengguna memiliki opsi untuk memasukkan kernel (dengan syarat harus ganjil) melalui halaman webnya yang akan digunakan untuk memproses fungsi tersebut.



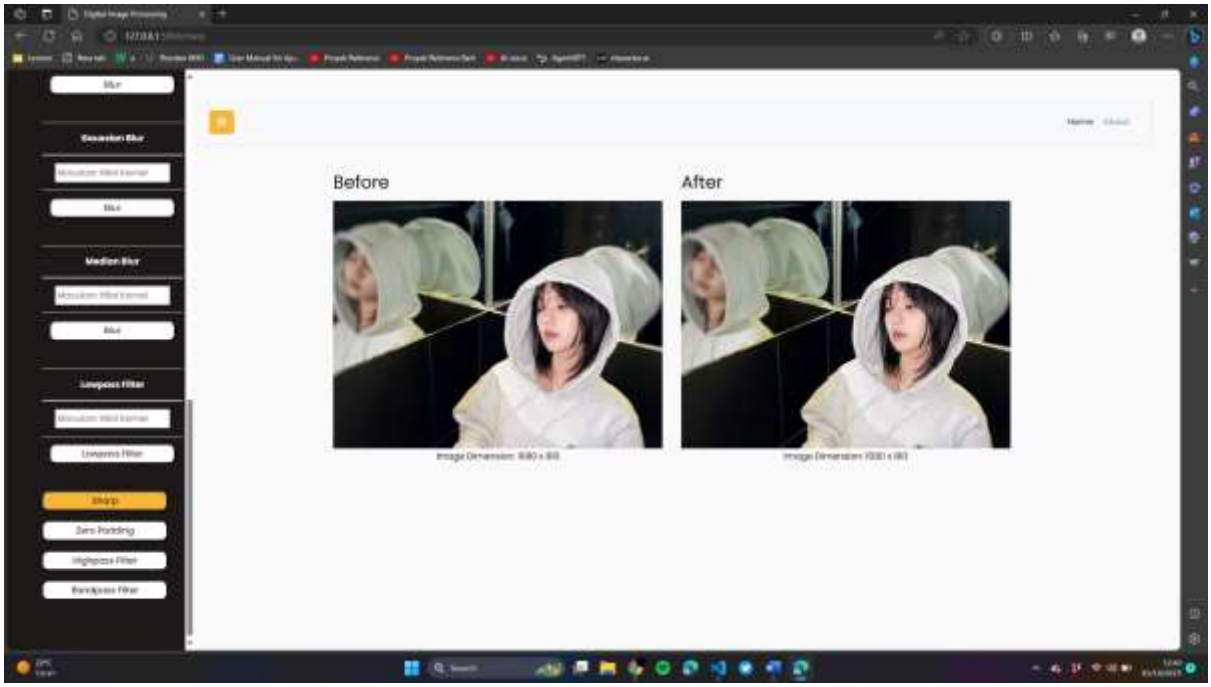
- Sharpening

```
def sharp():
    img = cv2.imread("static/img/img_normal.jpg")

    kernel = np.array([[0, -1, 0],
                       [-1, 5, -1],
                       [0, -1, 0]])

    sharp = cv2.filter2D(src=img, ddepth=-1, kernel=kernel)

    cv2.imwrite("static/img/img_now.jpg", sharp)
```

Fungsi sharp() memiliki tujuan untuk menerapkan efek penajaman (sharpening) pada gambar yang diberikan. Sharpening adalah teknik yang digunakan untuk meningkatkan kejelasan dan detail dalam gambar dengan menguatkan elemen-elemen penting sementara mengurangi detail-detail kecil dan noise.

```
kernel = np.array([[0, -1, 0],
                   [-1, 5, -1],
                   [0, -1, 0]])
```

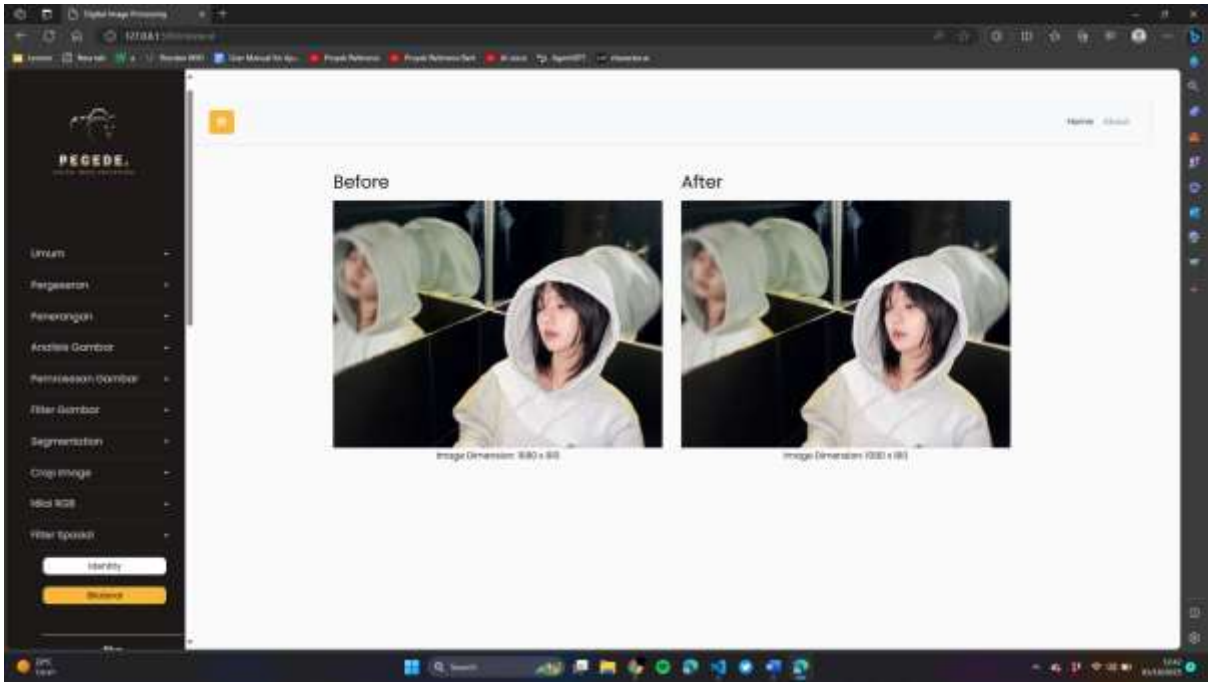
Kernel ini dibuat untuk melakukan operasi penajaman dengan metode peningkatan bobot pada piksel pusat (nilai 5) dan memberikan bobot negatif pada piksel-piksel di lingkungan sekitarnya. Ini bertujuan untuk meningkatkan kontrast antara nilai-nilai piksel dan menghasilkan efek penajaman.



- Bilateral Filter

```
def bilateral():
    img = cv2.imread("static/img/img_normal.jpg")

    bf = cv2.bilateralFilter(src=img,d=9,sigmaColor=75,sigmaSpace=75)

    cv2.imwrite("static/img/img_now.jpg", bf)
```

Fungsi bilateralFilter() digunakan untuk menerapkan filter bilateral pada gambar yang diberikan. Filter bilateral adalah jenis filter yang digunakan untuk mengurangi noise dalam gambar sambil tetap mempertahankan tepi dan detail yang penting. Filter ini mempertimbangkan dua faktor utama: perubahan nilai piksel dalam domain spasial dan perbedaan warna dalam domain warna.



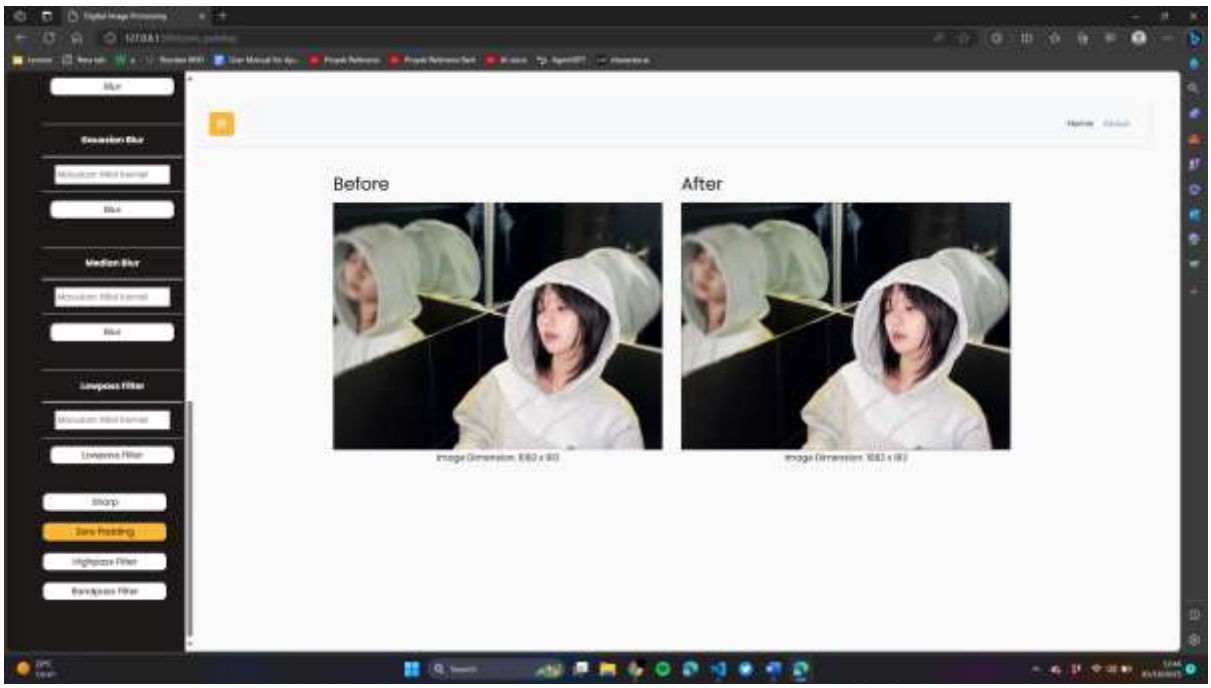
- Zero Padding

```
def zero_padding():
    img = cv2.imread("static/img/img_normal.jpg")

    image = cv2.copyMakeBorder(img, 1, 1, 1, 1, cv2.BORDER_CONSTANT, value=0)

    cv2.imwrite("static/img/img_now.jpg", image)
```

Fungsi zero_padding() memiliki tujuan untuk melakukan penambahan padding berisi nol pada gambar yang diberikan. Padding adalah langkah untuk menambahkan piksel tambahan di sekeliling tepi gambar, dan dalam konteks ini, padding tersebut berisi nilai nol.

```
image = cv2.copyMakeBorder(img, 1, 1, 1, 1, cv2.BORDER_CONSTANT, value=0)
```

1, 1, 1, 1: Angka-angka ini mengontrol lebar padding yang akan ditambahkan di bagian atas, bawah, kiri, dan kanan gambar. Di sini, kita menambahkan 1 piksel padding pada setiap sisi gambar. cv.BORDER_CONSTANT: Parameter ini menentukan jenis padding yang akan diterapkan, yaitu padding dengan nilai konstan. value=0: Parameter ini menetapkan nilai konstan yang akan digunakan untuk padding. Dalam kasus ini, kita menggunakan nilai 0, sehingga padding akan diisi dengan piksel berwarna hitam (nol).

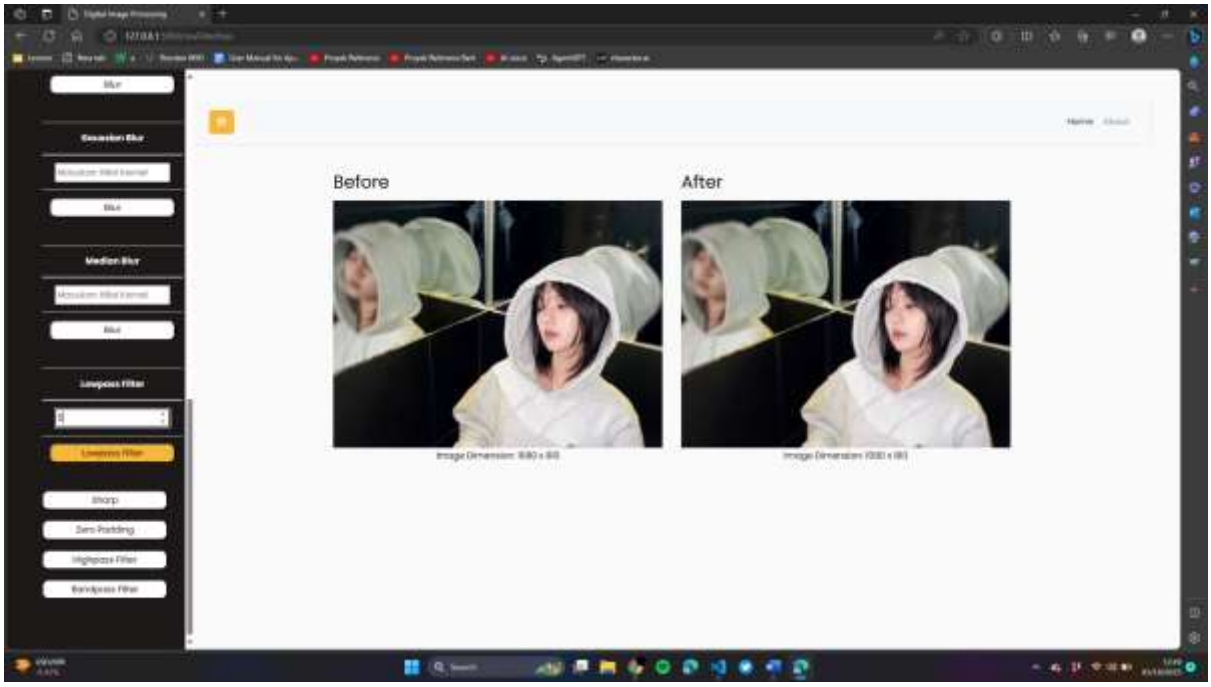


- Low Filter Pass

```
def lowFilterPass(kernel):
    img = cv2.imread("static/img/img_normal.jpg")
    # create the low pass filter
    lowFilter = np.ones((kernel,kernel),np.float32)/(kernel*kernel)
    # apply the low pass filter to the image
    lowFilterImage = cv2.filter2D(img,-1,lowFilter)

    cv2.imwrite("static/img/img_now.jpg", lowFilterImage)
```

Filter low-pass adalah jenis filter yang memungkinkan komponen frekuensi gambar yang memiliki nilai rendah untuk dilewati, sementara membatasi atau memblokir komponen frekuensi tinggi. Dengan kata lain, filter ini berfungsi untuk menghaluskan komponen gambar yang memiliki perubahan cepat atau tingkat detail yang tinggi. Ini dapat menghasilkan efek penyaringan yang mengurangi detail halus atau noise pada gambar.

Kemudian, filter low-pass dibentuk menggunakan fungsi np.ones() untuk menghasilkan matriks dengan dimensi (kernel, kernel), di mana semua elemennya diatur ke nilai 1. Ukuran filter ini ditentukan oleh parameter kernel yang diberikan kepada fungsi. Filter tersebut kemudian dinormalisasi dengan membaginya dengan jumlah total elemen dalam filter, yang sama dengan kernel * kernel. Hasilnya adalah filter yang digunakan untuk meredam atau melemahkan komponen frekuensi tinggi pada gambar. Dengan kata lain, filter ini membantu mengurangi tingkat detail atau perubahan cepat pada gambar. Pada fungsi Lowpass filter ini, pengguna dapat menginputkan nilai kernel melalui halaman webnya yang akan digunakan untuk memproses fungsi tersebut.



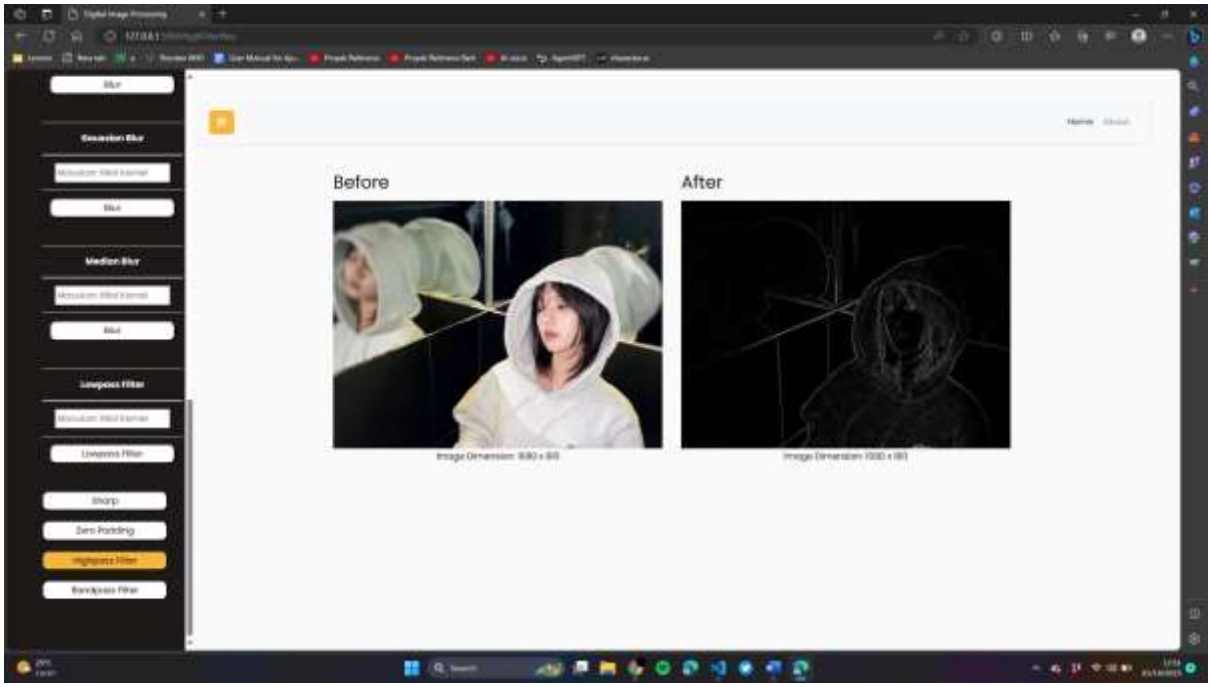
- High Filter Pass

```
def highFilterPass():
    img = cv2.imread("static/img/img_normal.jpg")
    # create the high pass filter
    highFilter = np.array([[-1,-1,-1],[-1,8,-1],[-1,-1,-1]])
    # apply the high pass filter to the image
    highFilterImage = cv2.filter2D(img,-1,highFilter)

    cv2.imwrite("static/img/img_now.jpg", highFilterImage)
```

Fungsi ini memiliki tujuan untuk menerapkan operasi filter high-pass pada gambar yang diberikan. Filter high-pass adalah jenis filter yang memungkinkan komponen frekuensi tinggi pada gambar untuk melewati filter, sementara membatasi atau memblokir komponen frekuensi rendahnya. Filter ini berdampak meningkatkan detail dan menonjolkan perubahan tajam dalam gambar.

```
highFilter = np.array([[-1,-1,-1],[-1,8,-1],[-1,-1,-1]])
```

Ini digunakan untuk mendefinisikan kernel highpass. Kernel ini adalah matriks yang akan digunakan untuk melakukan operasi konvolusi pada gambar. Kernel yang didefinisikan memiliki nilai-nilai spesifik yang dirancang untuk menciptakan efek highpass.



- Band Filter Pass

```
def bandFilterPass():
    img = cv2.imread("static/img/img_normal.jpg")
    # create the band pass filter
    bandFilter = np.array([[0,-1,0],[-1,4,-1],[0,-1,0]])
    # apply the band pass filter to the image
    bandFilterImage = cv2.filter2D(img,-1,bandFilter)

    cv2.imwrite("static/img/img_now.jpg", bandFilterImage)
```

Fungsi bandFilterPass digunakan untuk menerapkan filter bandpass pada gambar yang diambil dari file gambar yang terletak di "static/img/img_now.jpg". Filter bandpass ini bertujuan untuk menjaga komponen frekuensi dalam rentang tertentu sambil menghilangkan frekuensi yang terlalu rendah dan terlalu tinggi.

```
bandFilter = np.array([[0,-1,0],[-1,4,-1],[0,-1,0]])
```

Ini digunakan untuk mendefinisikan kernel bandpass. Kernel ini adalah matriks yang akan digunakan untuk melakukan operasi konvolusi pada gambar. Kernel yang didefinisikan memiliki nilai-nilai spesifik yang dirancang untuk menciptakan efek bandpass, dengan nilai-nilai negatif dan nol digunakan untuk mengurangi pengaruh frekuensi rendah dan tinggi.