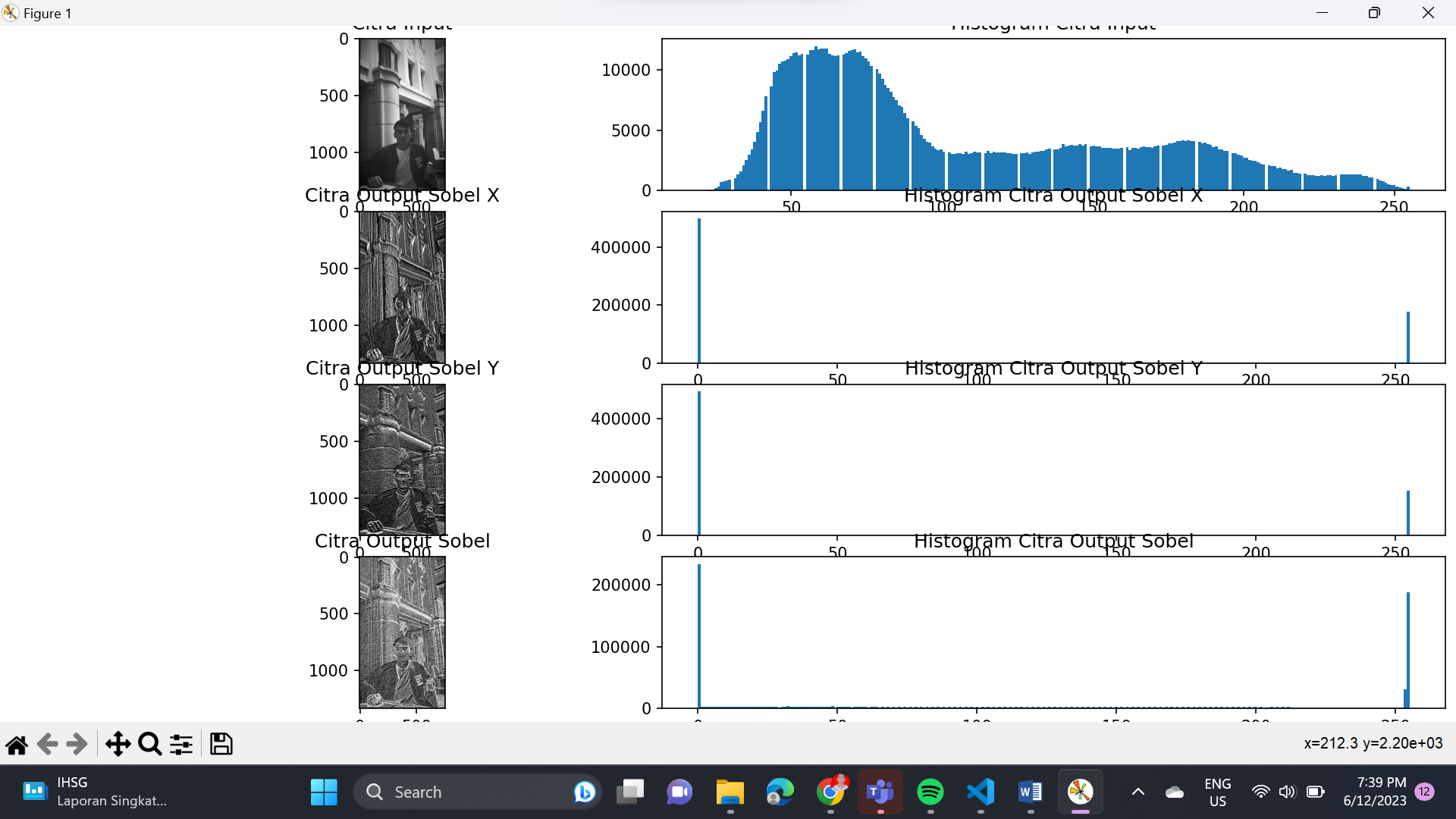
Nama : Reilham Riziq Algani

Nim : 1207070101

Kelas : Teknik Elektro C, TT

Praktukum 9

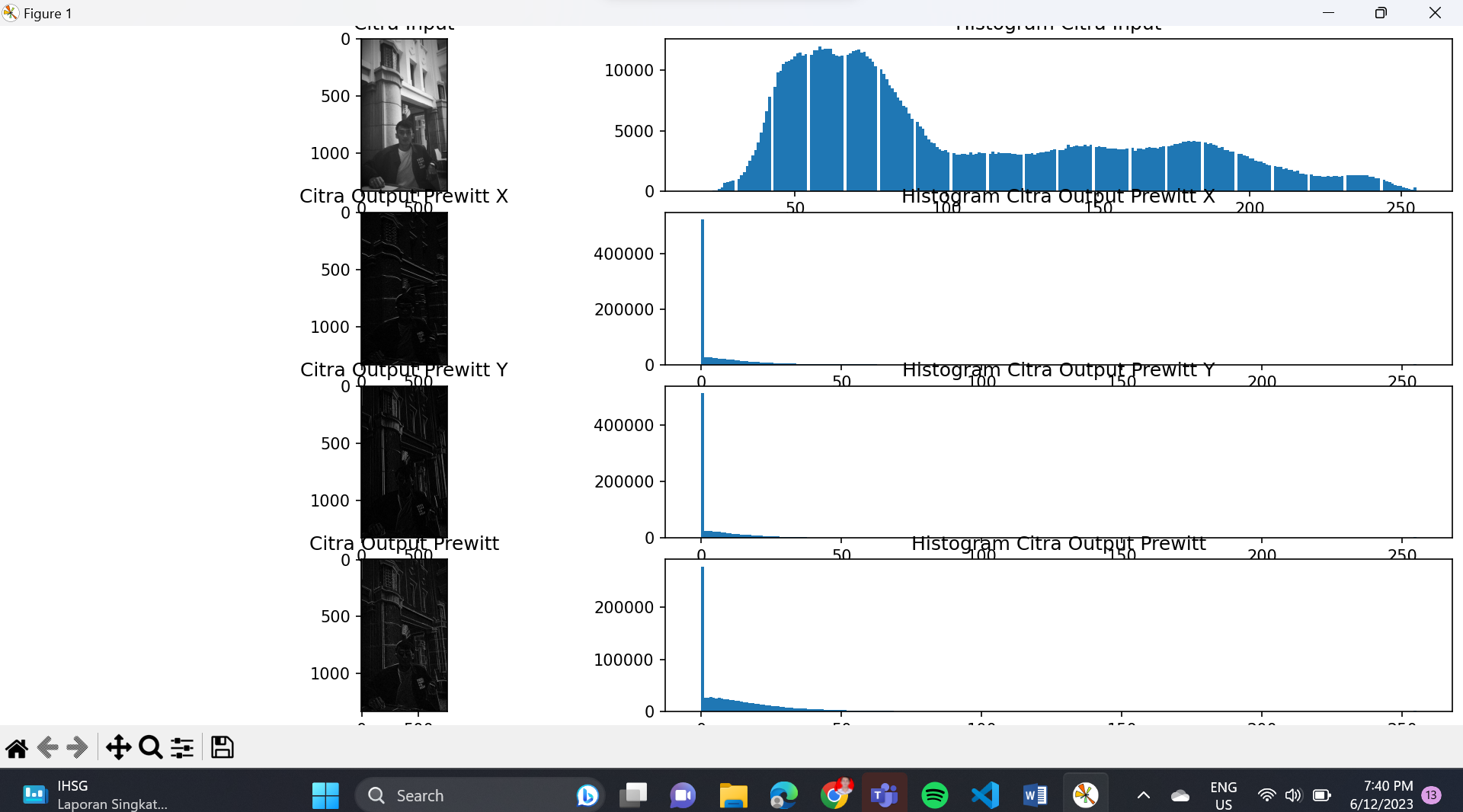
1. Praktikum 9.1 Sobel



Program ini menggunakan metode Sobel untuk deteksi tepi pada gambar. Proses dimulai dengan membaca gambar input menggunakan fungsi cv.imread() dengan mode cv.IMREAD\_GRAYSCALE untuk mengubahnya menjadi citra grayscale. Selanjutnya, metode Sobel diterapkan melalui fungsi cv.Sobel(). Dalam hal ini, Sobel diterapkan secara terpisah untuk menghitung gradien dalam arah horizontal (img\_sobelx) dan arah vertikal (img\_sobely). Setelah itu, kedua citra gradien digabungkan dengan menghitung jumlah intensitas piksel untuk membentuk citra deteksi tepi (img\_sobel). Selain itu, program ini menampilkan citra input, histogram citra input, citra deteksi tepi dalam arah X (img\_sobelx), histogram citra deteksi tepi dalam arah X, citra deteksi tepi dalam arah Y (img\_sobely), histogram citra deteksi tepi dalam arah Y, citra deteksi tepi (img\_sobel), dan histogram citra deteksi tepi. Histogram digunakan untuk visualisasi distribusi intensitas piksel dalam citra.

Hasilnya adalah citra yang menunjukkan lokasi tepi dengan intensitas yang lebih tinggi. Keunggulan metode Sobel adalah kesimpelannya dan kemudahannya dalam implementasi. Metode ini juga cenderung menghasilkan tepi yang tajam. Namun, kelemahan Sobel adalah ketidakmampuannya dalam mempertimbangkan arah tepi yang tegak lurus terhadap tepi sebenarnya.

1. Praktikum 9.2 prewitt



Program ini memanfaatkan metode Prewitt untuk deteksi tepi pada gambar. Langkah awalnya adalah membaca gambar input dan mengubahnya menjadi citra grayscale. Kemudian, metode Prewitt diterapkan pada citra grayscale menggunakan fungsi cv.filter2D(). Dalam proses ini, kernel Prewitt secara terpisah didefinisikan untuk menghitung gradien dalam arah horizontal (kernelx) dan arah vertikal (kernely). Hasil deteksi tepi dari kedua arah tersebut kemudian digabungkan dengan menjumlahkan intensitas piksel untuk membentuk citra hasil deteksi tepi (img\_prewitt). Program ini juga menampilkan citra input, histogram citra input, citra hasil deteksi tepi dalam arah X (img\_prewittx), histogram citra hasil deteksi tepi dalam arah X, citra hasil deteksi tepi dalam arah Y (img\_prewitty), histogram citra hasil deteksi tepi dalam arah Y, citra hasil deteksi tepi (img\_prewitt), dan histogram citra hasil deteksi tepi. Kelebihan dan kekurangan metode Prewitt serupa dengan metode Sobel.

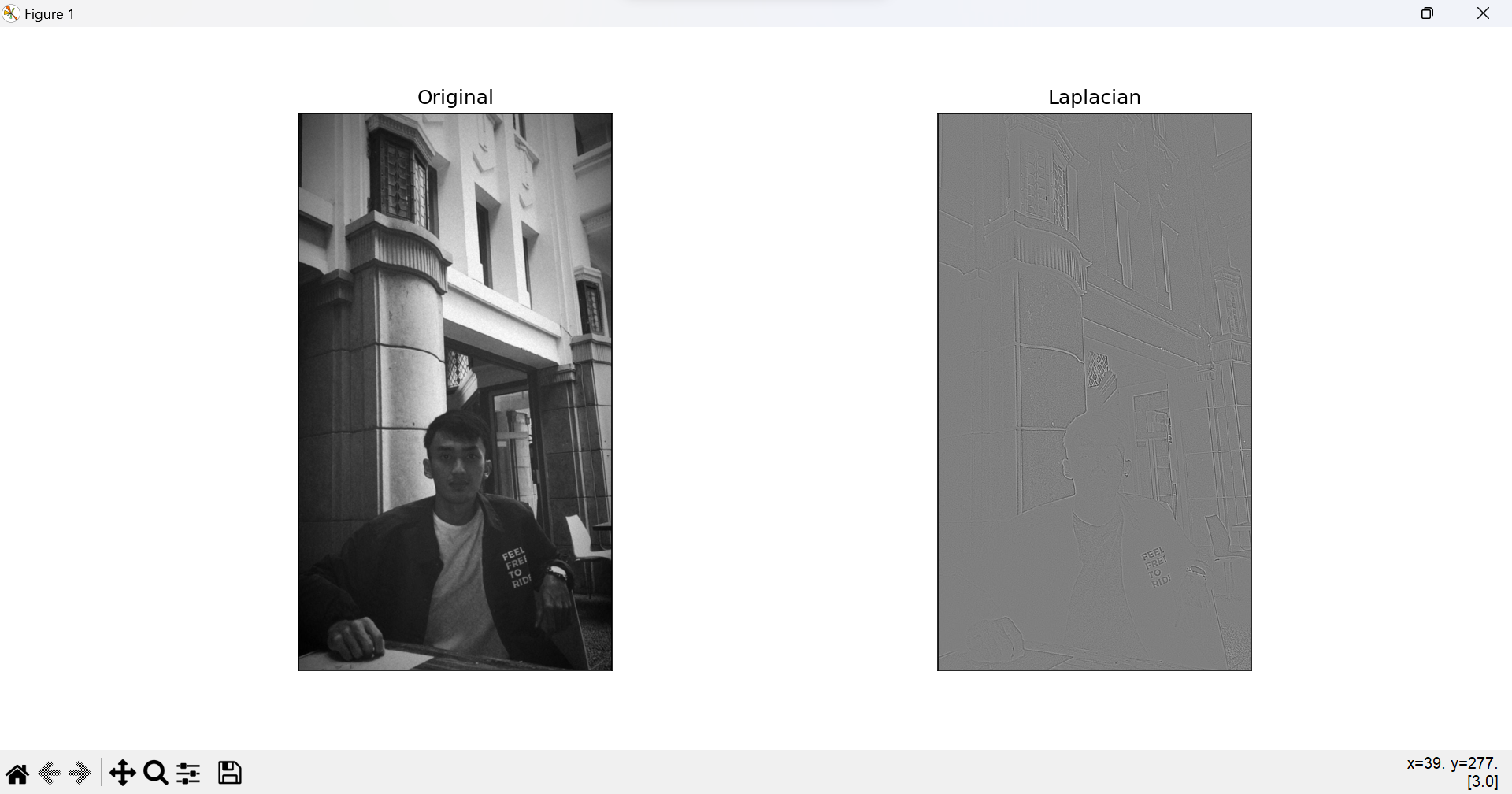
1. Praktikum 9.3 Canny



Program ini memanfaatkan metode Canny untuk deteksi tepi pada gambar. Proses dimulai dengan membaca gambar input dan mengubahnya menjadi citra grayscale. Metode Canny diterapkan pada citra grayscale menggunakan fungsi cv.Canny(). Dalam implementasi ini, nilai ambang bawah (threshold lower) dan ambang atas (threshold upper) diatur menjadi 100 dan 200. Hasil deteksi tepi disimpan dalam variabel img\_canny. Program ini juga menampilkan citra input, histogram citra input, citra hasil deteksi tepi (img\_canny), dan histogram citra hasil deteksi tepi.

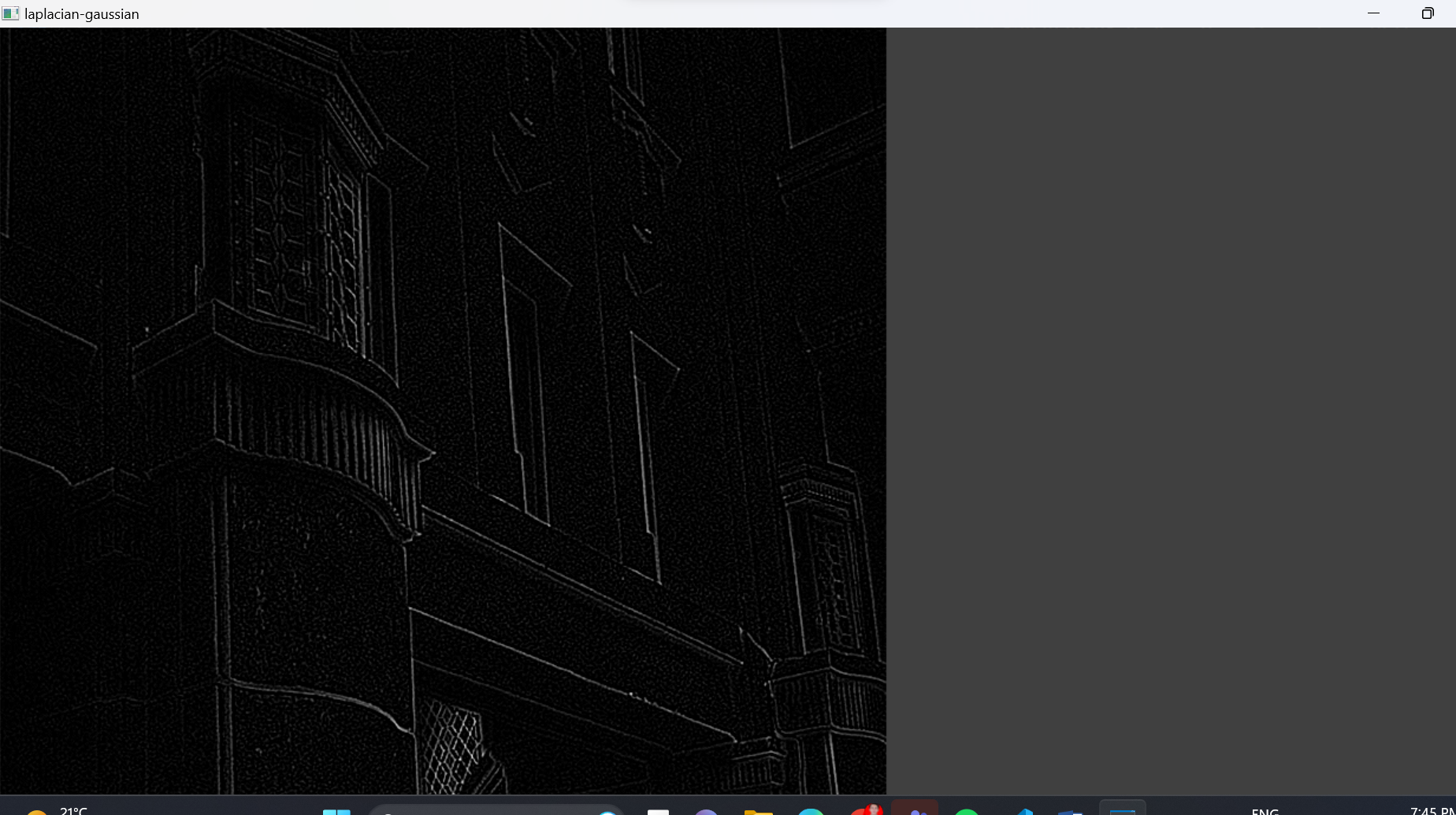
Hasilnya adalah citra yang menunjukkan tepi dengan intensitas yang tinggi. Kelebihan metode Canny adalah memberikan deteksi tepi yang akurat dan tajam. Metode ini juga dapat menghilangkan noise dan menghubungkan tepi yang terputus. Namun, kekurangan metode Canny adalah membutuhkan lebih banyak komputasi dan memiliki implementasi yang lebih kompleks.

1. Praktikum 9.4 Laplacian



Program ini memanfaatkan metode Laplacian untuk deteksi tepi pada gambar. Langkah awalnya adalah membaca gambar input dan mengonversinya menjadi citra grayscale. Untuk menghilangkan noise pada citra grayscale, program menggunakan filter Gaussian dengan ukuran kernel 3x3 melalui fungsi cv2.GaussianBlur(). Selanjutnya, metode Laplacian diterapkan pada citra yang telah terbebas dari noise menggunakan fungsi cv2.Laplacian() untuk mendeteksi perubahan intensitas piksel yang tajam. Hasil deteksi tepi (laplacian) ditampilkan menggunakan matplotlib. Kelebihan metode Laplacian adalah memberikan deteksi tepi yang tajam, terutama pada tepi yang memiliki perubahan intensitas yang drastis. Namun, kekurangan metode ini adalah sensitivitasnya terhadap noise pada gambar.

1. Praktikum 9.5 Laplacian 2



Program ini juga menggunakan metode Laplacian untuk mendeteksi tepi pada gambar. Langkah awalnya adalah membaca gambar input dan mengubahnya menjadi citra grayscale. Dilakukan penghilangan noise pada citra grayscale menggunakan filter Gaussian. Kemudian, metode Laplacian diterapkan pada citra hasil penghilangan noise menggunakan fungsi cv2.Laplacian(). Hasil deteksi tepi disimpan dalam variabel laplacian. Untuk memastikan hasil deteksi tepi dapat dengan mudah diinterpretasikan, hasilnya dinormalisasi dengan membaginya dengan nilai maksimum dari tepi (laplacian/laplacian.max()). Citra hasil deteksi tepi kemudian ditampilkan menggunakan OpenCV. Program ini menampilkan citra hasil deteksi tepi yang dihasilkan oleh metode Laplacian setelah penghilangan noise menggunakan filter Gaussian untuk mendeteksi perubahan intensitas piksel yang tajam. Kelebihan dan kekurangan metode Laplacian pada Program Laplacian 2 serupa dengan Program Laplacian 1.

Kesimpulan

Apabila menginginkan metode deteksi tepi yang akurat dan tajam dengan kemampuan menghilangkan noise dan menghubungkan tepi yang terputus, direkomendasikan untuk menggunakan metode Canny.

* metode yang sederhana dan mudah diimplementasikan, baik metode Sobel maupun Prewitt bisa menjadi pilihan yang baik.
* Metode Laplacian (baik Program Laplacian 1 maupun Laplacian 2) juga memberikan hasil yang tajam, namun perlu diperhatikan sensitivitasnya terhadap noise.
* Pemilihan metode terbaik tergantung pada kebutuhan spesifik dalam deteksi tepi, tingkat kompleksitas yang dapat ditoleransi, dan jenis gambar yang akan diproses.