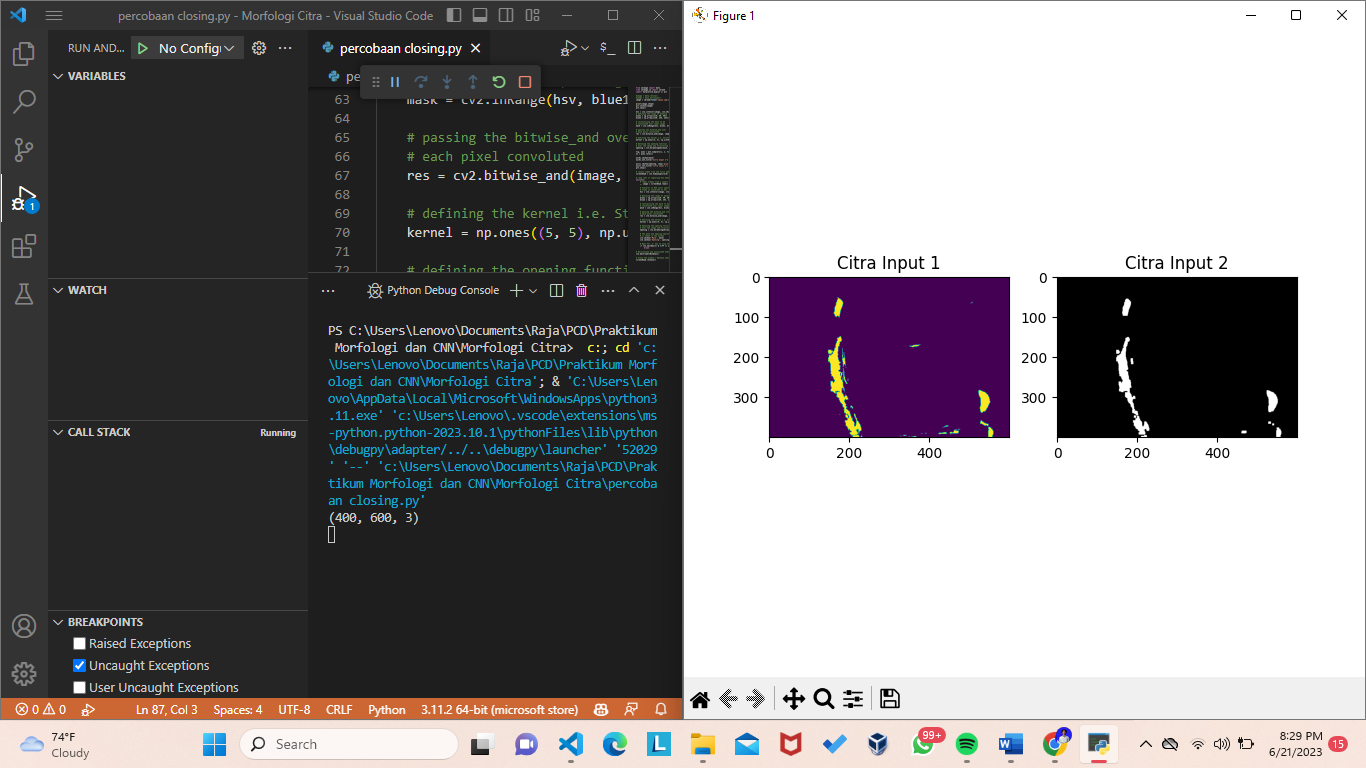
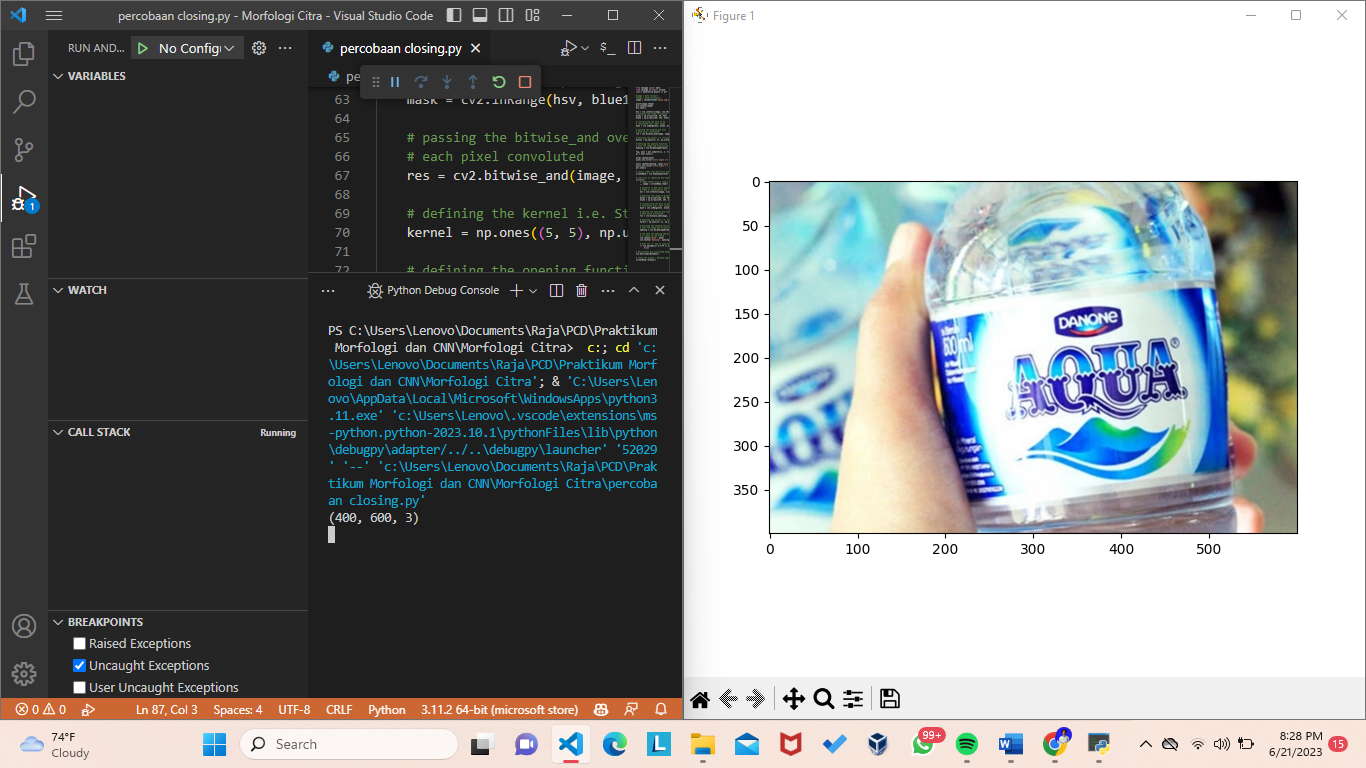
Nama : Rajasabian Ardhi Awangga

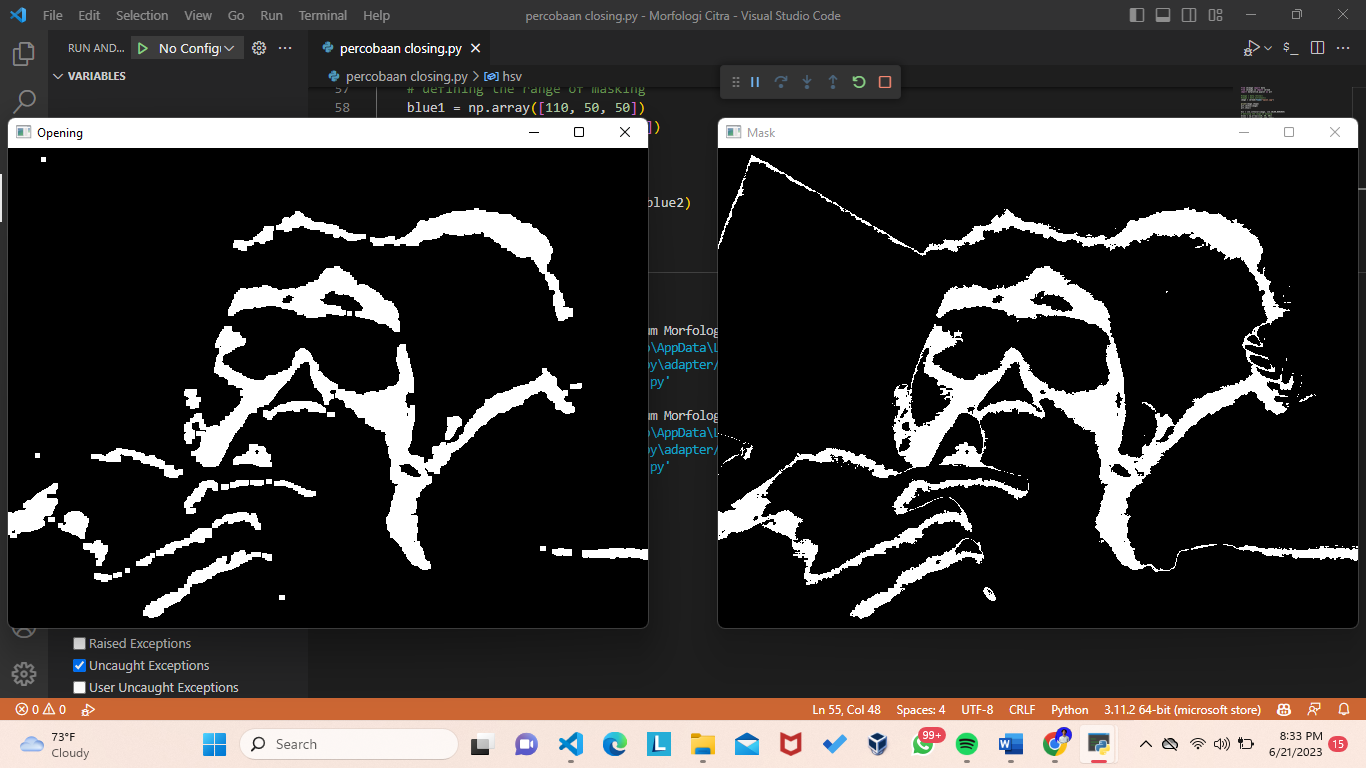
NIM : 1207070096

Mata Kuliah : Praktikum Pengolahan Citra Digital – TKK

PRAKTIKUM 11 Morfologi Citra

1. **Percobaan Closing**

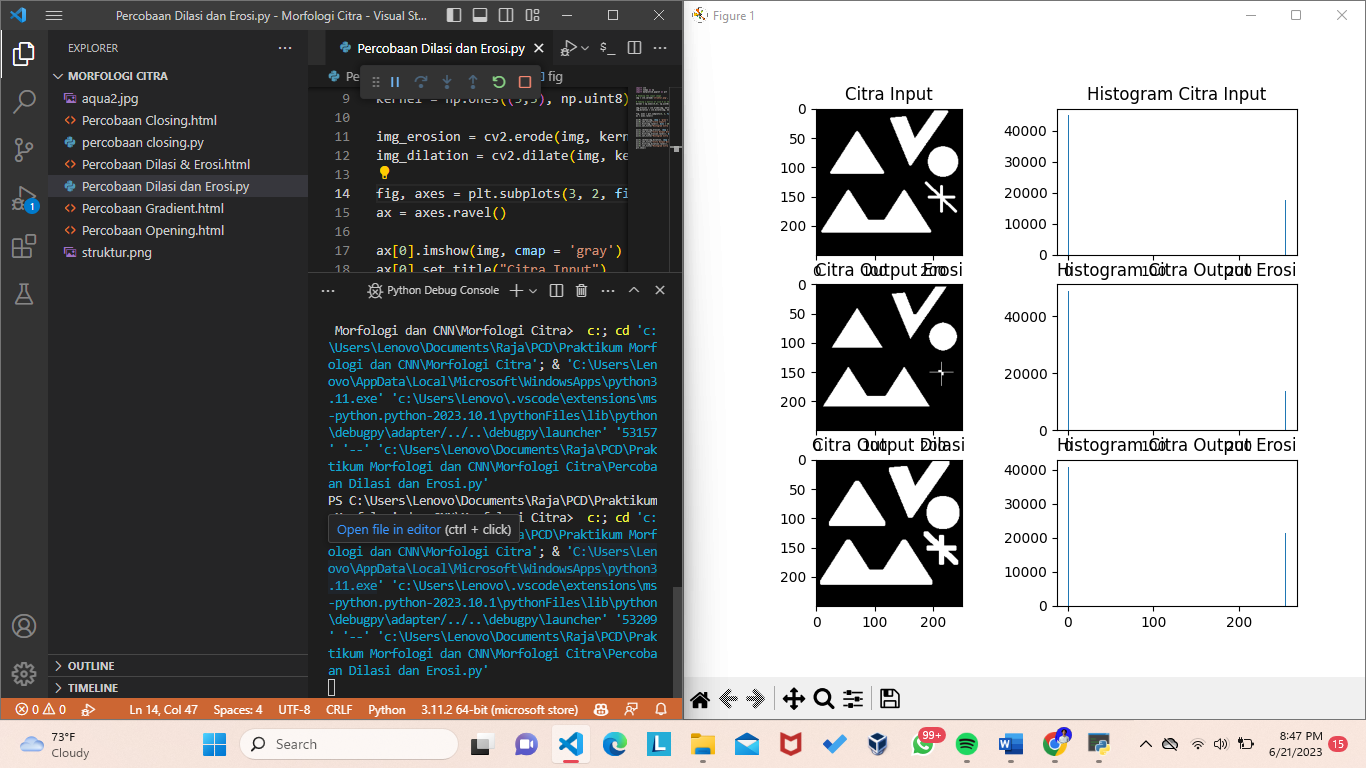




Closing merupakan operasi morfologi yang terdiri dari dua tahap, tetapi urutannya berbeda dari opening. Tahap pertama adalah dilasi diikuti oleh erosi pada citra. Operasi closing digunakan untuk mengisi lubang atau celah kecil dalam objek dan memperbaiki objek yang tidak tertutup sempurna. Operasi ini juga berguna untuk menghaluskan tepi objek yang ada. Pada hasil closing, lubang kecil dalam objek akan tertutup dan objek yang lebih kecil dari elemen struktur atau elemen struktur yang tidak lengkap akan diisi.

Pada Percobaan kali ini kodingan mengambil citra dari file dan juga memproses video dari webcam dengan menggunakan operasi morfologi opening. Hasilnya, ditampilkan mask yang berfungsi untuk mengidentifikasi area yang memiliki warna biru dan juga hasil dari operasi opening pada citra tersebut.

1. **Percobaan Dilasi dan Erosi**

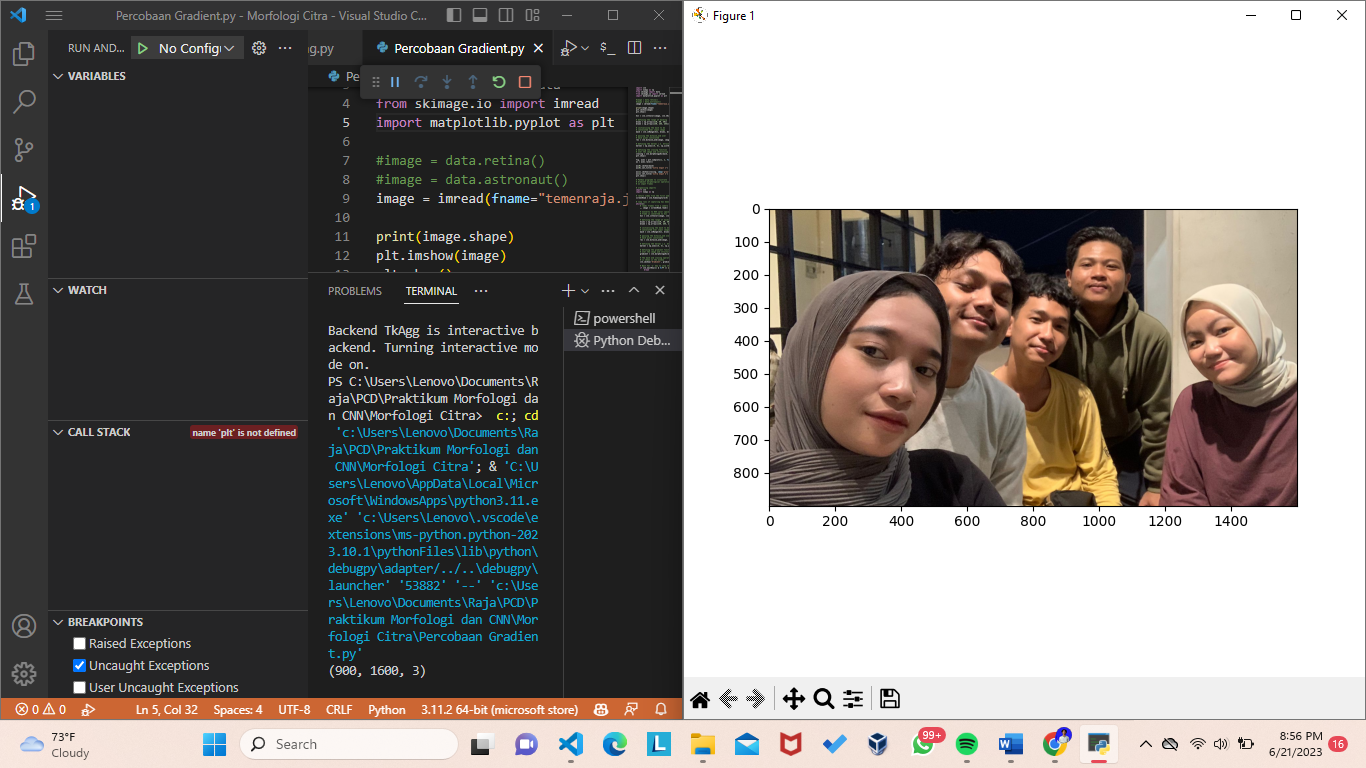


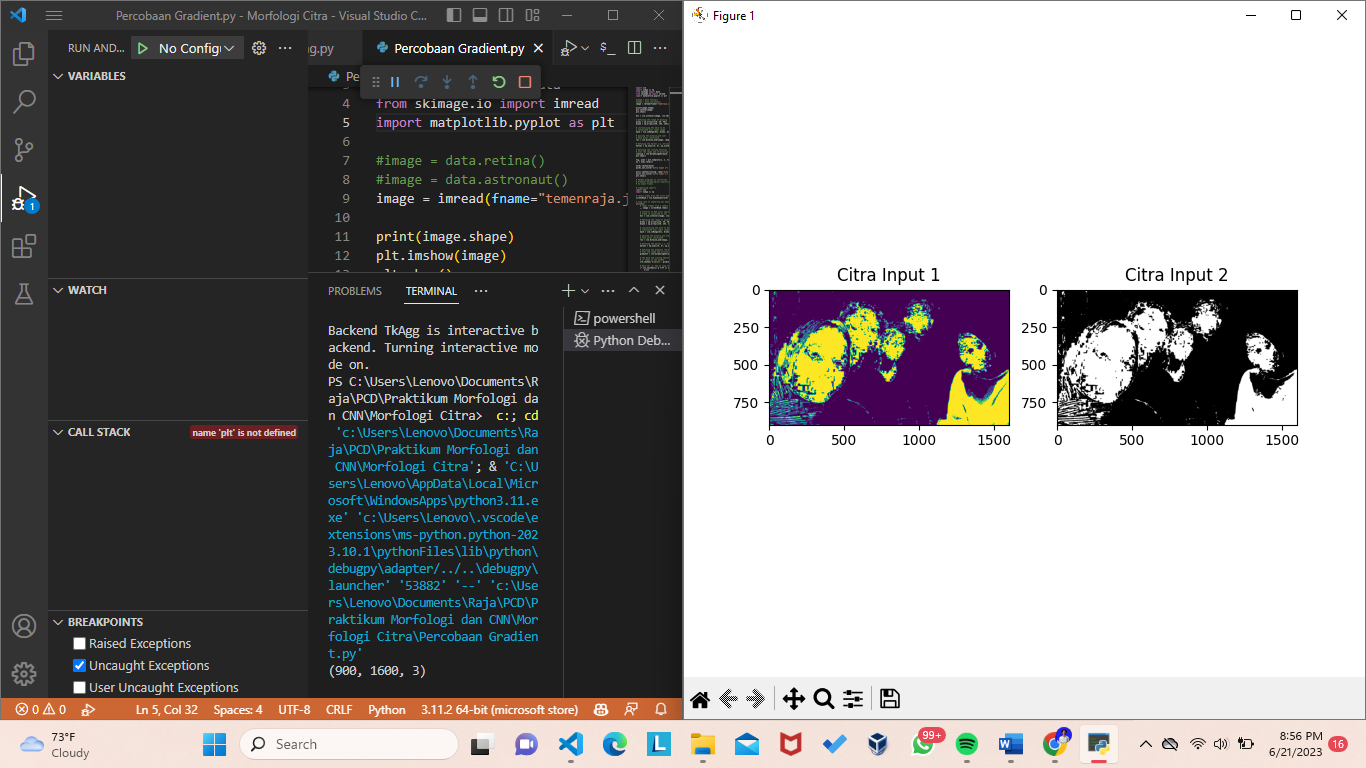
Dilasi adalah operasi yang digunakan untuk memperluas atau memperbesar objek dalam citra. Sedangkan, Erosi adalah operasi yang digunakan untuk mengikis atau menyusutkan objek dalam citra.

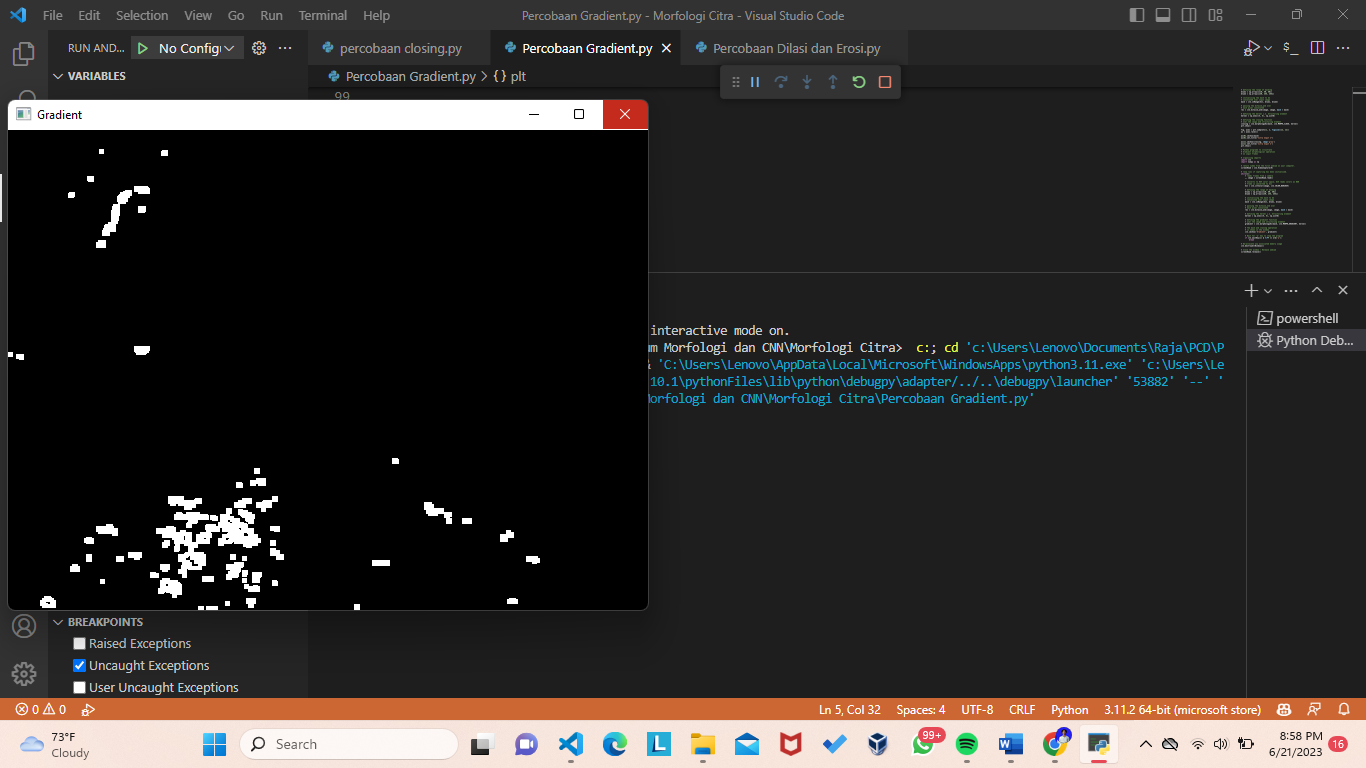
Dilasi dan erosi sering digunakan bersama-sama dalam rangkaian operasi morfologi citra. Misalnya, operasi opening adalah kombinasi erosi diikuti oleh dilasi, sedangkan operasi closing adalah kombinasi dilasi diikuti oleh erosi. Keduanya sangat berguna dalam menghilangkan noise, mengisi lubang, mengidentifikasi objek, dan memodifikasi bentuk objek dalam citra.

Pada percobaan ini kodingan mengambil citra dalam mode grayscale, kemudian melakukan erosi dan dilasi menggunakan kernel 5x5. Hasil erosi dan dilasi ditampilkan dalam bentuk citra, serta histogram dari citra input, citra hasil erosi, dan citra hasil dilasi juga ditampilkan. Hal ini membantu untuk memvisualisasikan perubahan dalam distribusi intensitas piksel setelah dilakukan operasi morfologi.

1. **Percobaan Gradient**



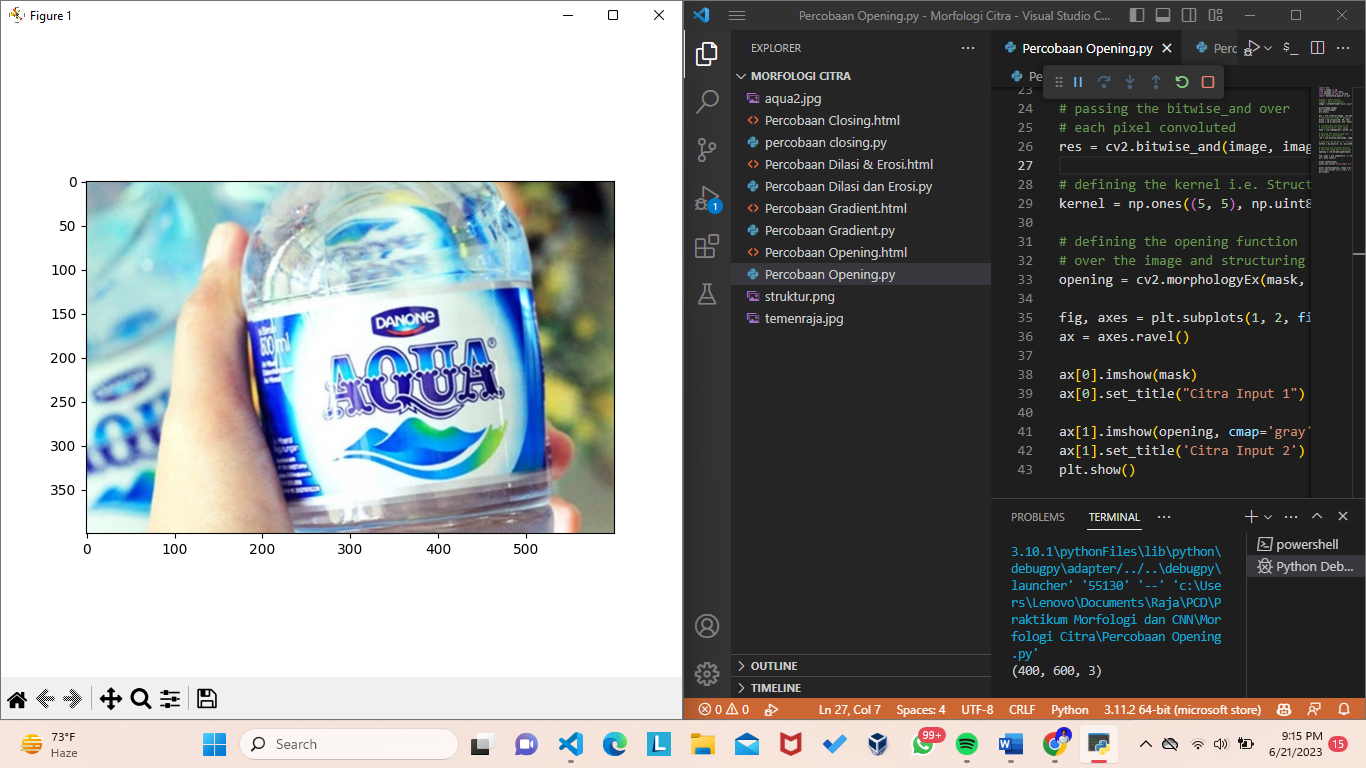


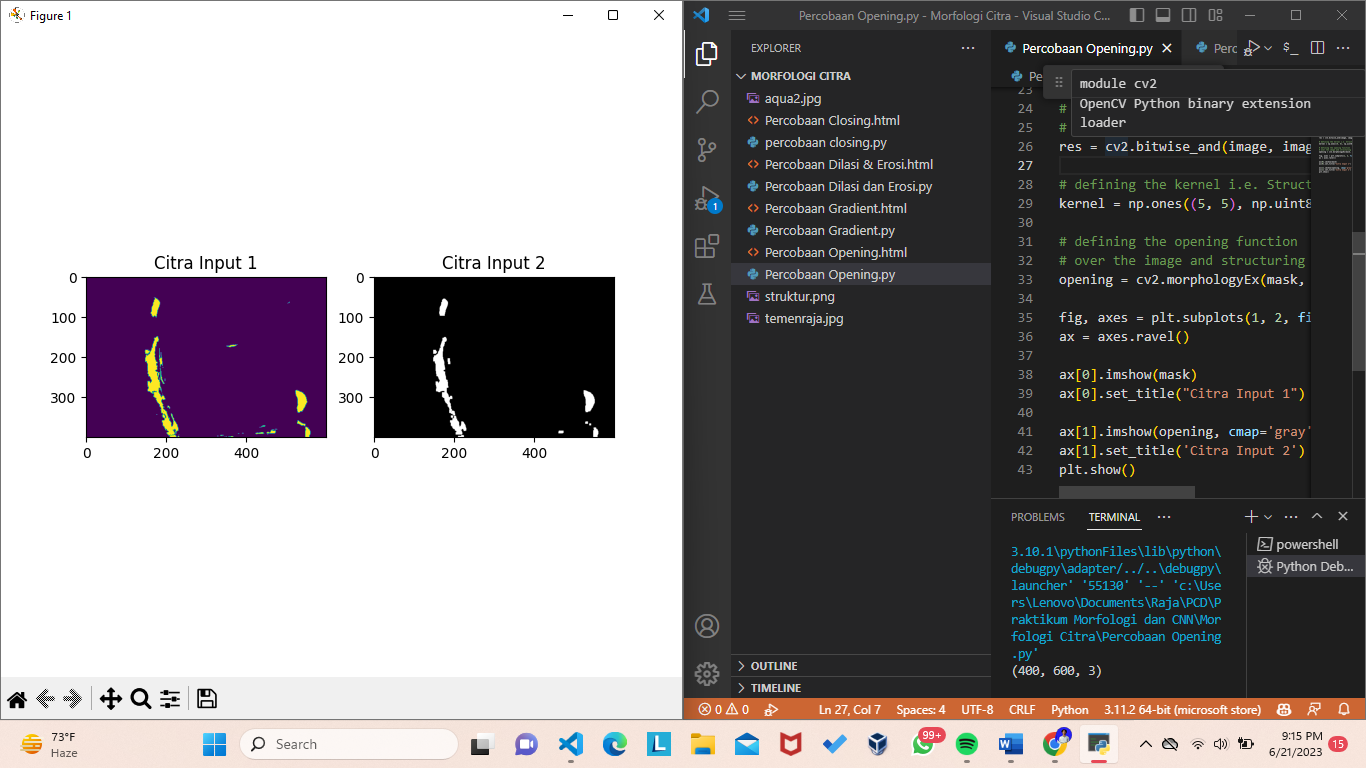


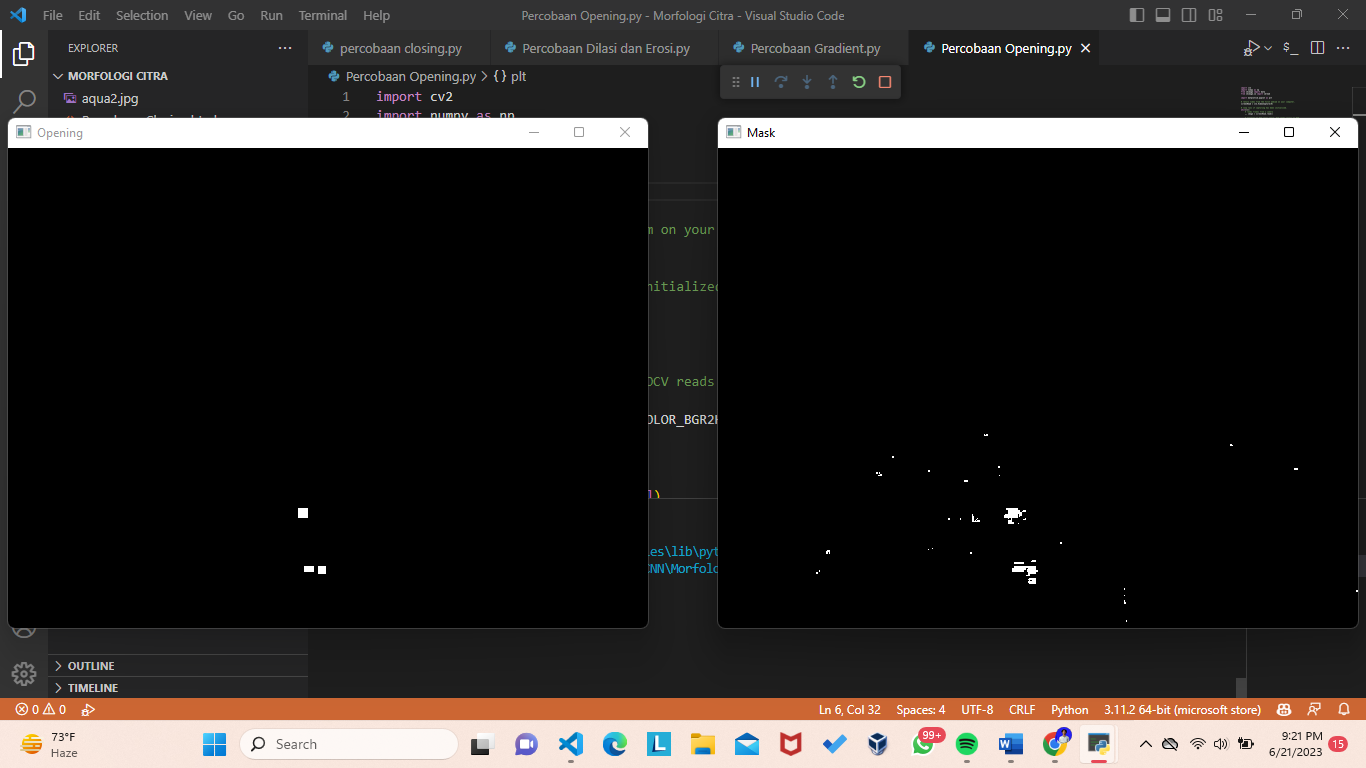
1. Import pustaka dan modul:
   * **cv2**: Pustaka utama untuk pemrosesan citra menggunakan OpenCV.
   * **numpy as np**: Pustaka untuk operasi numerik pada array dan matriks.
   * **skimage**: Pustaka untuk pemrosesan citra menggunakan scikit-image.
   * **imread** dari **skimage.io**: Modul untuk membaca citra.
   * **matplotlib.pyplot as plt**: Modul untuk menampilkan citra.
2. Membaca citra:
   * Citra dibaca menggunakan **imread()** dengan menggunakan file "temenraja.jpg" dan disimpan dalam variabel **image**.
3. Konversi warna citra:
   * Citra dikonversi dari BGR (Blue-Green-Red) ke HSV (Hue-Saturation-Value) menggunakan **cv2.cvtColor()** dan disimpan dalam variabel **hsv**.
4. Pembuatan mask:
   * Definisikan rentang warna biru dalam HSV sebagai **blue1** (nilai bawah) dan **blue2** (nilai atas).
   * Buat mask dengan menggunakan **cv2.inRange()** dengan memasukkan **hsv**, **blue1**, dan **blue2**.
   * Mask digunakan untuk memfilter citra dengan menggunakan **cv2.bitwise\_and()**, dan hasilnya disimpan dalam variabel **res**.
5. Morfologi operasi "closing":
   * Didefinisikan kernel dengan matriks 5x5 menggunakan **np.ones()**.
   * Operasi "closing" dilakukan menggunakan **cv2.morphologyEx()** dengan citra mask, operasi MORPH\_CLOSE, dan kernel.
   * Hasil "closing" disimpan dalam variabel **closing**.
6. Menampilkan citra:
   * Menggunakan **subplots()** untuk membuat satu baris dan dua kolom dalam satu figure.
   * Menggunakan **imshow()** untuk menampilkan citra mask pada **ax[0]** dan citra hasil "closing" pada **ax[1]**.
   * Memberikan judul pada masing-masing citra.
7. Menampilkan figure:
   * Menggunakan **show()** untuk menampilkan figure yang berisi citra.
8. Operasi "gradient" pada citra dari webcam:
   * Membuka webcam menggunakan **cv2.VideoCapture()**.
   * Dalam loop while, membaca frame dari webcam menggunakan **screenRead.read()**.
   * Mengkonversi frame ke ruang warna HSV menggunakan **cv2.cvtColor()**.
   * Menggunakan **cv2.inRange()** dan **cv2.bitwise\_and()** untuk membuat mask dan mengaplikasikannya pada frame.
   * Operasi "gradient" dilakukan menggunakan **cv2.morphologyEx()** dengan citra mask, operasi MORPH\_GRADIENT, dan kernel.
   * Hasil "gradient" ditampilkan menggunakan **cv2.imshow()**.
9. Menghentikan program:
   * Jika tombol 'a' ditekan, loop while akan berhenti dan program ditutup menggunakan **cv2.destroyAllWindows()** dan **screenRead.release()**.

Kodingan ini membaca citra "temenraja.jpg", menampilkan citra mask dan citra hasil "closing". Selain itu, menggunakan webcam untuk mengambil frame dan menampilkan citra hasil "gradient".

1. **Percobaan Opening**







Kodingan ini mirip dengan kodingan sebelumnya, tetapi dengan penggunaan operasi morfologi "opening" pada citra.

1. Import pustaka dan modul:
   * Pustaka dan modul yang diimpor sama dengan sebelumnya.
2. Membaca citra:
   * Citra dibaca menggunakan **imread()** dengan menggunakan file "aqua2.jpg" dan disimpan dalam variabel **image**.
3. Konversi warna citra:
   * Citra dikonversi dari BGR (Blue-Green-Red) ke HSV (Hue-Saturation-Value) menggunakan **cv2.cvtColor()** dan disimpan dalam variabel **hsv**.
4. Pembuatan mask:
   * Definisikan rentang warna biru dalam HSV sebagai **blue1** (nilai bawah) dan **blue2** (nilai atas).
   * Buat mask dengan menggunakan **cv2.inRange()** dengan memasukkan **hsv**, **blue1**, dan **blue2**.
   * Mask digunakan untuk memfilter citra dengan menggunakan **cv2.bitwise\_and()**, dan hasilnya disimpan dalam variabel **res**.
5. Morfologi operasi "opening":
   * Didefinisikan kernel dengan matriks 5x5 menggunakan **np.ones()**.
   * Operasi "opening" dilakukan menggunakan **cv2.morphologyEx()** dengan citra mask, operasi MORPH\_OPEN, dan kernel.
   * Hasil "opening" disimpan dalam variabel **opening**.
6. Menampilkan citra:
   * Menggunakan **subplots()** untuk membuat satu baris dan dua kolom dalam satu figure.
   * Menggunakan **imshow()** untuk menampilkan citra mask pada **ax[0]** dan citra hasil "opening" pada **ax[1]**.
   * Memberikan judul pada masing-masing citra.
7. Menampilkan figure:
   * Menggunakan **show()** untuk menampilkan figure yang berisi citra.
8. Operasi "opening" pada citra dari webcam:
   * Membuka webcam menggunakan **cv2.VideoCapture()**.
   * Dalam loop while, membaca frame dari webcam menggunakan **screenRead.read()**.
   * Mengkonversi frame ke ruang warna HSV menggunakan **cv2.cvtColor()**.
   * Menggunakan **cv2.inRange()** dan **cv2.bitwise\_and()** untuk membuat mask dan mengaplikasikannya pada frame.
   * Operasi "opening" dilakukan menggunakan **cv2.morphologyEx()** dengan citra mask, operasi MORPH\_OPEN, dan kernel.
   * Hasil "opening" ditampilkan menggunakan **cv2.imshow()**.
9. Menghentikan program:
   * Jika tombol 'a' ditekan, loop while akan berhenti dan program ditutup menggunakan **cv2.destroyAllWindows()** dan **screenRead.release()**.

Kodingan ini membaca citra "aqua2.jpg", menampilkan citra mask dan citra hasil "opening". Selain itu, menggunakan webcam untuk mengambil frame dan menampilkan citra hasil "opening" secara real-time.

**KESIMPULAN**

Dalam kesimpulan pada percobaan kali ini, opening dan closing digunakan untuk mengubah bentuk objek dan menghilangkan detail yang tidak diinginkan dalam citra, sedangkan gradient digunakan untuk menyoroti perbedaan intensitas dan tepi objek. Dilasi dan erosi sangat berguna dalam menghilangkan noise, mengisi lubang, mengidentifikasi objek, dan memodifikasi bentuk objek dalam citra.