Adi Rizka (G64150013)

|  |
| --- |
| **import** **cv2** **import** **numpy** **as** **np** |

Meng-*import library* opencv dan *library* numpy

|  |
| --- |
| image = cv2.imread("LennaInput.jpg", **0**) output = cv2.imread("LennaOutput.jpg", **0**) row, col = image.shape |

Membuka gambar LennaInput.jpg dan LennaOutput.jpg sebagai grayscale menggunakan fungsi *imread* yang tersedia di *library* opencv. Kemudian mengambil jumlah baris dan kolom dengan menggunakan fungsi *shape*.

|  |
| --- |
| **def** **BlurMedian**(image, size): |

Mendefinisikan fungsi *BlurMedian* yang dapat menerima argumen berupa gambar *input* dan ukuran *kernel* yang di inginkan.

|  |
| --- |
| dst = np.zeros((row, col, **1**), np.uint8) matrix = size batas = matrix // **2** |

Mendefinisikan kanvas yang akan digunakan yaitu variabel dst menggunakan library *numpy* yang diisi dengan angka nol. Kemudian memindahkan argumen *size* ke variabel *matrix* . Setelah itu memberikan batas atas dan batas bawah dari *matrix-size* yang didapatkan dari argumen. *Matrix-size* dibagi 2 karena *pixel* yang akan digunakan adalah *pixel* tengah sehingga ukuran *matrix* harus dipisahkan untuk *matrix* atas dan *matrix* bawah.

|  |
| --- |
| **for** i **in** range(row):  **for** j **in** range(col):  **if** i >= batas **and** i <= **255**-batas **and** j >= batas **and** j <= **255**-batas: |

Meng-iterasi tiap baris dan kolom tapi kolom yang diproses berbeda tergantung dengan ukuran *kernel* yang diinginkan. Jadi program tidak akan meng-akses array yang tidak tersedia atau di luar dari 0 - 255.

|  |
| --- |
| vect = [] **for** k **in** range(i-batas, i+batas+**1**):  **for** l **in** range(j-batas, j+batas+**1**):  vect.append(image[k][l]) vect.sort() |

Mendefinisikan *list* dengan nama variabel *vect*. Kemudian melakukan iterasi dengan ukuran *matrix* yang diinginkan. Awal *pixel* yang akan diproses adalah baris, kolom *pixel* di kurangi batas dan baris, kolom *pixel* ditambah batas dan di tambah 1 karena *for* pada python memiliki kondisi kurang dari bukan kurang dari sama dengan. Kemudian nilai tiap pixel tadi dimasukkan ke *list vect*. Kemudian kita melakukan sort pada *list* agar bisa mendapatkan median dari *matrix kernel* tersebut.

|  |
| --- |
| mid = matrix\*matrix // **2** dst.itemset((i, j, **0**), vect[mid]) |

Mendapatkan nilai mid yaitu posisi median yang akan dipakai. Kemudian memasukkan nilai ke variabel dst yang merupakan kanvas yang telah didefinisikan terlebih dahulu menggunakan numpy.

|  |
| --- |
| **return** dst |

Mengembalikan hasil gambar yang telah di proses.

|  |
| --- |
| destination = BlurMedian(image, **3**) destination = BlurMedian(destination, **3**) |

Melakukan *medianBlur* pada gambar kemudian memasukkan ke variabel *destination.* Kemudian karena 2 kali proses *filtering* dengan *kernel* 3 menghasilkan gambar yang paling dekat dengan *output* yang diminta, dilakukan *medianBlur* sekali lagi.

|  |
| --- |
| cv2.imshow("My Lenna", destination) cv2.imshow("Lenna Expectation", output) cv2.waitKey() cv2.destroyAllWindows() cv2.destroyAllWindows() |

Menampilkan gambar *Lena* hasil proses dengan *kernel* 3 dan 2 kali proses *filtering* dan gambar *Lena* target. Kemudian mendapatkan waitKey() agar gambar dapat tertutup setelah menerima *input* dari *Keyboard.* Kemudian men-*destroy* semua jendela gambar yang terbuka.