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

Meng-*import* *library* yang dibutuhkan untuk pengolahan gambar yaitu *numpy* dan *cv2* (*OpenCV*).

|  |
| --- |
| cameraman = cv2.imread('k\_cameraman.jpg', **0**) equalized = cv2.imread('k\_equalized.jpg', **0**)  row, col = cameraman.shape rowe, cole = equalized.shape |

Membuka gambar *k\_cameraman.jpg* dan *k\_equalized.jpg* sebagai grayscal dengan menambah argumen ‘0’. Kemudian mendapatkan jumlah baris dan jumlah kolom tiap pixel yang ada pada kedua gambar.

|  |
| --- |
| rata, rata2 = **0**, **0** **for** i **in** range(row):  **for** j **in** range(col):  rata += cameraman[i,j]  rata2 += equalized[i,j]  rata /= (row\*col) rata2 /= (row\*col) |

Menginisiasi variabel rata dan rata2 dengan nilai awal 0. Variabel ini akan digunakan untuk menampung nilai rata-rata dari tiap *pixel* pada gambar *k\_cameraman.jpg* dan *k\_equalized.jpg*. Kemudian melakukan *looping* pada setiap *pixel* pada kedua gambar tersebut sementara variabel rata dan rata2 terus bertambah dengan nilai total dari tiap pixel gambar. Setelah didapatkan nilai total pixel gambar *k\_cameraman.jpg* dan *k\_equalized.jpg* kemudian variabel rata dan rata2 dibagi dengan total *pixel* yang ada pada kedua gambar yaitu baris dikali kolom.

|  |
| --- |
| new\_image\_c = np.zeros((row, col, **1**), np.uint8) new\_image\_e = np.zeros((rowe, cole, **1**), np.uint8) |

Membuat kanvas baru untuk hasil olahan gambar *k\_cameraman.jpg* dan *k\_equalized.jpg* dengan nama *new\_image\_c* dan *new\_image\_e*.

|  |
| --- |
| **for** i **in** range(row):  **for** j **in** range(col):  new\_image\_c[i, j] = cameraman[i, j]\***0.5** **if** cameraman[i,j] < rata **else** **2**\*cameraman[i, j]  new\_image\_e[i, j] = equalized[i, j]\***0.5** **if** equalized[i,j] < rata2 **else** **2**\*equalized[i, j] |

Melakukan *looping* pada tiap *pixel* gambar *cameraman* dan *equalized*. Nilai suatu *pixel* dikali 0.5 jika nilainya kurang dari rataan dan dikali 2 jika lebih besar dari atau sama dengan nilai rataannya.

|  |
| --- |
| final\_image = np.zeros((row, col, **1**), np.uint8) |

Membuat kanvas untuk hasil akhir gambar.

|  |
| --- |
| **for** i **in** range(row):  **for** j **in** range(col):  res = **0** **if** new\_image\_e[i,j] == new\_image\_c[i,j] **else** new\_image\_e[i,j]  final\_image.itemset((i, j, **0**), res) |

Melakukan looping pada tiap pixel yang ada pada gambar *k\_cameraman.jpg (new\_image\_c)* dan *k\_equalized.jpg (new\_image\_e)* dan memasukkan nilai 0 jika nilai *pixel* pada *new\_image\_c* dan *new\_image\_e* sama, namun jika tidak sama variabel res akan dimasukkan nilai pixel *new\_image\_c*.

|  |
| --- |
| cv2.imshow('Final', final\_image) cv2.imshow('Cameraman', cameraman) cv2.imshow('Equalized', equalized) cv2.waitKey(**0**) cv2.destroyAllWindows() |

Menampilkan gambar hasil operasi yang telah dilakukan sebelumnya yaitu *final\_image*. Kemudian juga menampilkan gambar k\_*cameraman.jpg* dan gambar *k\_equalized.jpg* sebagai perbandingan. Kemudian menambahkan waitKey(0) agar gambar tidak langsung tertutup dan menambahkan destroyAllWindows() agar semua windows tertutup setelah program tertutup.