**Wingatun S Tartiana (G64150050)**

1. **Kode Python**

**import** cv2  
**import** numpy **as** np  
  
image = cv2.imread(**"car.png"**, 0)  
  
**def** grayscale(source):  
 row, col, ch = source.shape  
 gray\_kanvas = np.zeros((row, col, 1), np.uint8)  
 **for** i **in** range(0, row):  
 **for** j **in** range(0, col):  
 blue, green, red = source[i, j]  
 gray = red \* 0.299 + green \* 0.587 + blue \* 0.114  
 gray\_kanvas.itemset((i, j, 0), gray)  
 **return** gray\_kanvas  
  
**def** Stretching(gambar):  
 row, col = gambar.shape  
 output = np.zeros((row, col, 1), np.uint8)  
 min = max = gambar[0, 0]  
  
 **for** i **in** range(0, row):  
 **for** j **in** range(0, col):  
 **if** gambar[i, j] < min:  
 min = gambar[i, j]  
 **if** gambar[i, j] > max:  
 max = gambar[i, j]  
  
 **for** i **in** range(0, row):  
 **for** j **in** range(0, col):  
 normalize = (float(gambar[i, j] - min) / (max - min)) \* 255  
 output.itemset((i, j, 0), normalize)  
 **return** output  
  
**def** histogramgray(gambar):  
 buckets = [0] \* 300  
 arraynorm = [0] \* 300  
 histogramcol = 255  
 histogramrow = 150  
 border = 30  
 kanvashistogram = np.zeros(((histogramrow + border), histogramcol, 1), np.uint8)  
 row, col = gambar.shape  
 gray\_kanvas = np.zeros((row, col, 1), np.uint8)  
 **for** i **in** range(0, row):  
 **for** j **in** range(0, col):  
 buckets[int(gambar[i, j])] += 1  
 maks = max(buckets)  
 mins = min(buckets)  
 **for** intent **in** range(0, 255):  
 jumlahperbar = buckets[intent]  
 normal = int(float(jumlahperbar) / float(maks) \* float(histogramrow))  
 arraynorm[intent] = normal  
 **for** y **in** range(int(histogramrow - normal + border), histogramrow + border):  
 kanvashistogram.itemset((y, intent, 0), 255)  
 **return** kanvashistogram  
  
**def** histogramgrayRaw(gambar):  
 buckets = [0] \* 300  
 arraynorm = [0] \* 300  
 histogramcol = 255  
 histogramrow = 150  
 border = 30  
 kanvashistogram = np.zeros(((histogramrow + border), histogramcol, 1), np.uint8)  
 row, col, raw = gambar.shape  
 gray\_kanvas = np.zeros((row, col, 1), np.uint8)  
 **for** i **in** range(0, row):  
 **for** j **in** range(0, col):  
 buckets[int(gambar[i, j])] += 1  
 maks = max(buckets)  
 mins = min(buckets)  
 **for** intent **in** range(0, 255):  
 jumlahperbar = buckets[intent]  
 normal = int(float(jumlahperbar) / float(maks) \* float(histogramrow))  
 arraynorm[intent] = normal  
 **for** y **in** range(int(histogramrow - normal + border), histogramrow + border):  
 kanvashistogram.itemset((y, intent, 0), 255)  
 **return** kanvashistogram  
  
**def** equalization(gambar):  
 row, col = gambar.shape  
 canvas = np.zeros((row, col, 1), np.uint8)  
  
 *#kemunculan tiap pixel* pixel = [0] \* 256  
 **for** i **in** range(0, row):  
 **for** j **in** range(0, col):  
 nilai = int(gambar[i, j])  
 pixel[nilai] += 1  
  
 *#peluang nilai kemunculan tiap pixel* **for** i **in** range(0, 256):  
 pixel[i] = float(pixel[i]) / float(row \* col)  
  
 *#histogramgram kumulatif* **for** i **in** range(0, 256):  
 pixel[i] = pixel[i] + pixel[i - 1]  
  
 *#equalized histogramgram* **for** i **in** range(0, 256):  
 pixel[i] = pixel[i] \* (256 - 1)  
  
 *#masukkan ke dalam kanvas* **for** i **in** range(0, row):  
 **for** j **in** range(0, col):  
 nilai = int(gambar[i, j])  
 nilai = pixel[nilai]  
 canvas.itemset((i, j, 0), nilai)  
  
 **return** canvas  
  
image\_contrast = Stretching(image)  
image\_equalized = equalization(image)  
  
cv2.imshow(**"Asli"**, image)  
cv2.imshow(**"histogram Asli"**, histogramgray(image))  
  
cv2.imshow(**"Stretching"**, image\_contrast)  
cv2.imshow(**"histogram Stretching"**, histogramgrayRaw(image\_contrast))  
  
cv2.imshow(**"Equalized"**, image\_equalized)  
cv2.imshow(**"histogram Equalized"**, histogramgrayRaw(image\_equalized))  
  
cv2.waitKey(0)  
cv2.destroyAllWindows()

1. Hasil

