|  |  |  |
| --- | --- | --- |
|  | **Lembar Kerja Mahasiswa**  **Mata Kuliah Pengolahan Citra Digital Praktik (203311-20)**  **Program Studi Informatika**  **Fakultas Sains & Teknologi – Universitas Teknologi Yogyakarta** | |
|  | **Identitas Mahasiswa** | |
| **Nama** | **Herlambang Kurniawan** |
| **NPM** | **5200411434** |
| **Kelompok Prak** | **Kelompok 1** |
|  | | |
| **Soal 1.** | | |
| Berdasarkan demo di kelas, buatlah aplikasi berbasis GUI untuk salah satu dari 3 operasi dasar yang disampaikan :   1. Operasi Negative citra 2. OperasiPenjumlahan dan pengurangan Citra dengan Citra 3. Operasi Perkalian dan pembagian citra dengan skalar   Untuk tugas kali ini Anda dapat membuat layout GUI Anda sendiri. Buatlah layout GUI yang menarik dan tetap mudah digunakan. | | |
| **Hasil Script** | | |
| **//tuliskan script python Anda di sini**  from tkinter import \* import numpy as np from tkinter import filedialog import cv2 as cv from PIL import Image, ImageTk import os from ttkbootstrap import Style from tkinter import ttk  # MEMBUAT SEBUAH CLASS UNTUK WINDOWS\_1 class Window\_1:  # MEMBUAT CONSTRUCTOR CLASSNYA  def \_\_init\_\_(self, window\_def = None):  # MENGATUR AGARA KETIKA PARAMETER windows\_def ITU TIDAK ADA MAKA PASS SAJA  if window\_def is None :  pass  # MENGATUR JIKA PARAMETER WINDOW\_DEF ADA ISINYA MAKA DESTROY  else :  window\_def.destroy()   # MENGATUR SIZE DARI WINDOWNYA DAN STYLE DARI BOOSTRAPNYA  self.window = Tk()  self.window.title("Image Browse App - 5200411434")  self.window.geometry("1305x650+25+25")  self.window.config(bg="#323232")  style = Style(theme='darkly')   # MEMBUAT LEFT , MIDDLE , RIGHT FRAME  self.left\_frame = Frame(self.window, width=500, height=700, bg='#323232')  self.left\_frame.grid(row=0, column=0, padx=10, pady=5,sticky='n')   self.middle\_frame = Frame(self.window, width=600, height=550, bg='#323232')  self.middle\_frame.grid(row=0, column=1, padx=10, pady=5,sticky='n')   self.right\_frame = Frame(self.window, width=100, height=550, bg='#222222')  self.right\_frame.grid(row=0, column=2, padx=5, pady=15,sticky='n')   # BAGIAN LEFTFRAME AKAN DIBUAT PREVIEW GAMBAR DAN BUTTON BUAT MENGAMBIL GAMBAR DARI STORAGE ,FILTER ,MENU DAN EXIT  self.gambar = Label(self.left\_frame,height=13,width=25)  self.gambar.grid(row=0,column=0,pady=0)   # BUTTON AMBIL GAMBAR  ttk.Style().configure('Outline.TButton',bordercolor="none",foreground="white",background="#2F8886",width=15, font=('Helvetica',10))  self.icon\_picture = PhotoImage(file="resources/icon/picture\_icon.png")  self.search = ttk.Button(self.left\_frame,text=" Image Browser",image=self.icon\_picture,style="Outline.TButton",command=self.get\_image,compound="left")  self.search.grid(row=1, column=0, padx=5, pady=0,ipadx=20)   # RADIO BUTTON BUAT BAGIAN FILTERNYA  self.tool\_bar\_2 = Frame(self.left\_frame, width=180, height=200, bg="#3f3f3f")  self.tool\_bar\_2.grid(row=2, column=0, padx=10, pady=50, sticky='w')   self.icon\_filter = PhotoImage(file="resources/icon/filter\_icon.png")  self.filter = Label(self.tool\_bar\_2,image=self.icon\_filter,text="Filter",compound='left',bg="#FD7014")  self.filter.grid(row=0,column=0,ipadx=55,columnspan=2)   # RB BAGIAN GRAYSCALE  ttk.Style().configure('info.TRadiobutton', background="#3f3f3f", foreground='white', font=('Helvetica', 8))  self.btn\_to\_gray = ttk.Radiobutton(self.tool\_bar\_2,text="Grayscale",value=1,style='info.TRadiobutton',command=self.rgb\_to\_grayscale)  self.btn\_to\_gray.grid(row=1, column=0, padx=25, pady=10, sticky='w')   # RB BAGIAN BLUE CHANNEL  self.btn\_to\_blue = ttk.Radiobutton(self.tool\_bar\_2, text="Blue Channel", value=2, style='info.TRadiobutton',command=lambda:self.take\_channel("blue"))  self.btn\_to\_blue.grid(row=1, column=1, padx=25, pady=10, sticky='w')   # RB BAGIAN ORIGINAL IMAGE  self.btn\_to\_normal = ttk.Radiobutton(self.tool\_bar\_2, text="Original",value=3,style='info.TRadiobutton',command=self.img\_to\_normal)  self.btn\_to\_normal.grid(row=2, column=0, pady=10, padx=25, sticky='w')   # RB BAGIAN GREEN CHANNEL  self.btn\_to\_green = ttk.Radiobutton(self.tool\_bar\_2, text="Green Channel", value=4, style='info.TRadiobutton',command=lambda:self.take\_channel("green"))  self.btn\_to\_green.grid(row=2, column=1, padx=25, pady=10, sticky='w')   # RB BAGIAN NEGATIVE IMAGE  self.btn\_negative = ttk.Radiobutton(self.tool\_bar\_2, text="Negative Image",value=5,style='info.TRadiobutton',command=self.image\_negative)  self.btn\_negative.grid(row=3, column=0, pady=10, padx=25, sticky='w')   # RB BAGIAN RED CHANNEL  self.btn\_to\_red = ttk.Radiobutton(self.tool\_bar\_2, text="Red Channel", value=6, style='info.TRadiobutton',command=lambda:self.take\_channel("red"))  self.btn\_to\_red.grid(row=3, column=1, padx=25, pady=10, sticky='w')   #===========================================================================================================================   # MEMBUAT FRAME DI DALAM LEFT FRAME  self.tool\_bar\_3 = Frame(self.left\_frame, width=180, height=200, bg="#323232")  self.tool\_bar\_3.grid(row=4, column=0, padx=10, pady=10, sticky='w')   # BUTTON UNTUK GANTI MENU  self.btn\_window\_2 = Button(self.tool\_bar\_3, text="MENU", width=20, height=2, bd=0, bg="#57CC99", relief="solid",command=lambda:Window\_2(self.window))  self.btn\_window\_2.grid(row=4, column=0,pady=5,padx=5)   # BUTTON UNTUK EXIT  self.btn\_exit = Button(self.tool\_bar\_3, text="Exit", width=20, height=2, bd=0, bg="#da4453", relief="solid",command=lambda: exit())  self.btn\_exit.grid(row=5, column=0, pady=5, padx=5)  #===================================================================================================  # PADA BAGIAN RIGHT FRAME AKAN DIGUNAKAN UNTUK SLIDER THRESHOLD ,MULTIPLY DAN DIVIDE   # MEMBUAT SLIDER THRESHOLD  # LABEL THRESHOLD  self.label\_treshold = Label(self.right\_frame, text="Treshold", width=10, height=1, bd=0, bg="#5ea880",relief="solid")  self.label\_treshold.grid(row=0, column=0, pady=4, sticky='n',ipady=3)   # SLIDER  current\_value = IntVar()  self.scale = ttk.Scale(self.right\_frame,variable=current\_value, from\_=0, to=255, orient=HORIZONTAL, length=250,style="success.Horizontal.TScale",command=self.treshold)  self.scale.grid(row=1, column=0, pady=4, sticky='w')   # PARAMETER  self.label\_parameter = Label(self.right\_frame, text="0")  self.label\_parameter.grid(row=1, column=1, pady=4, sticky='w')  # ============================================================================================================  # MEMBUAT SLIDER MULTIPLY  # LABEL MULTIPLY  self.label\_multiply = Label(self.right\_frame, text="Multiply", width=10, height=1, bd=0, bg="#e74c3c",relief="solid")  self.label\_multiply.grid(row=2, column=0, pady=4, sticky='n',ipady=3)   # SLIDER  current\_value\_multiply = IntVar()  self.scale\_multiply = ttk.Scale(self.right\_frame, from\_=1, to=10, orient=HORIZONTAL,length=250,style="danger.Horizontal.TScale",variable=current\_value\_multiply,command=self.multiply)  self.scale\_multiply.grid(row=3, column=0, pady=4, sticky='w')   # PARAMETER  self.label\_parameter\_multiply = Label(self.right\_frame, text="1.00")  self.label\_parameter\_multiply.grid(row=3, column=1, pady=4, sticky='w')  # ============================================================================================================  # MEMBUAT SLIDER DIVIDE  self.label\_divide = Label(self.right\_frame, text="Divide", width=10, height=1, bd=0, bg="#3498db",relief="solid")  self.label\_divide.grid(row=4, column=0, pady=4, sticky='n',ipady=3)   #SLIDER  current\_value\_divide = IntVar()  self.scale\_divide = ttk.Scale(self.right\_frame, from\_=1, to=10, orient=HORIZONTAL, length=250,style="info.Horizontal.TScale", variable=current\_value\_divide,command=self.divide)  self.scale\_divide.grid(row=5, column=0, pady=4, sticky='w')   # PARAMETER  self.label\_parameter\_divide = Label(self.right\_frame, text="1.00")  self.label\_parameter\_divide.grid(row=5, column=1, pady=4, sticky='w') # ============================================================================================================   # BAGIAN UNTUK MENAMPILKAN DEFAULT IMAGE (JIKA GAMBAR BELUM DIPILIH)  self.image = Image.open("resources/noimage.jpg")   # image preview sebelah kiri  self.image\_l = self.image.resize((185, 185), Image.ANTIALIAS)  self.image\_l = ImageTk.PhotoImage(self.image\_l)  self.gambar = Label(self.left\_frame, image=self.image\_l)  self.gambar.grid(row=0, column=0, padx=5, pady=0)   # image preview tengah  self.image\_r = self.image.resize((650, 600), Image.ANTIALIAS)  self.image\_r = ImageTk.PhotoImage(self.image\_r)  self.gambar\_2 = Label(self.middle\_frame, image=self.image\_r)  self.gambar\_2.grid(row=0, column=0, padx=5, pady=10)    # METHOD UNTUK MENGAMBIL GAMBAR  def get\_image(self):  self.fln = filedialog.askopenfilename(initialdir=os.getcwd(), title="Select Image File",  filetypes=(("PNG Files", "\*.png"), ("JPG File", "\*.jpg"),  ("All Files", "\*.\*")))   self.file\_location = np.copy(self.fln)  self.image = Image.open(self.fln)  self.mini\_preview\_ori(self.image)  self.preview\_img(self.image)   # METHOD UNTUK MENAMPILKAN GAMBAR ORIGINAL  def mini\_preview\_ori(self,image):  self.image\_l = image.resize((185, 185), Image.ANTIALIAS)  self.image\_l = ImageTk.PhotoImage(self.image\_l)  self.gambar\_1 = Label(self.left\_frame,image=self.image\_l)  self.gambar\_1.grid(row=0, column=0, pady=20)   # METHOD UNTUK MENAMPILKAN GAMBAR HASIL FILTER  def preview\_img(self,image):  self.image\_r = image.resize((650, 600), Image.ANTIALIAS)  self.image\_r = ImageTk.PhotoImage(self.image\_r)  self.gambar\_2 = Label(self.middle\_frame, image=self.image\_r)  self.gambar\_2.grid(row=0, column=0, padx=5, pady=10)   # METHOD UNTUK MENGHILANGKAN SEMUA FILTER YANG ADA PADA GAMBAR  def img\_to\_normal(self):  img = Image.open(self.fln)   self.scale.set(0)  self.label\_parameter.config(text='0')   self.scale\_multiply.set(0)  self.label\_parameter\_multiply.config(text='1.00')   self.scale\_divide.set(0)  self.label\_parameter\_divide.config(text='1.00')   self.preview\_img(img)   # METHOD UNTUK FILTER GAMBAR YANG HANYA MENGAMBIL CHANNELL WARNA SAJA  def take\_channel(self,color):  img = cv.imread(str(self.file\_location))  img\_rgb = cv.cvtColor(img, cv.COLOR\_BGR2RGB)  (R,G,B) = cv.split(img\_rgb)  if color == "red":  R = Image.fromarray(R)  self.preview\_img(R)  elif color == "green":  G = Image.fromarray(G)  self.preview\_img(G)  elif color == "blue":  B = Image.fromarray(B)  self.preview\_img(B)   # METHOD UNTUK FILTER GRAYSCALE  def rgb\_to\_grayscale(self):  self.img = cv.imread(str(self.file\_location))  grayscale = np.zeros(self.img.shape)   R = self.img[:, :, 0]  G = self.img[:, :, 1]  B = self.img[:, :, 2]   R = R \* 0.299  G = G \* 0.587  B = B \* 0.114   total = R + G + B  grayscale = self.img.copy()   for i in range(3):  grayscale[:, :, i] = total   gray = Image.fromarray(grayscale)  self.preview\_img(gray)   # MENTHOD UNTUK MENGUBAH NEGATIVE IMAGE  def image\_negative(self):  img\_bgr = cv.imread(self.fln)  img\_rgb = cv.cvtColor(img\_bgr,cv.COLOR\_BGR2RGB)  img\_neg = 1 - img\_rgb  img\_negative = Image.fromarray(img\_neg)  self.preview\_img(img\_negative)   # METHOD UNTUK MENGATUR THRESHOLD  def treshold(self,event):  try:  self.label\_parameter.config(text=str(int(self.scale.get())))  img = cv.imread(str(self.file\_location), 0)  treshold = np.zeros(img.shape)   for i in range(img.shape[0]):  for j in range(img.shape[1]):  pixel = img[i, j]  if pixel < int(self.scale.get()):  treshold[i, j] = 0 \* 255  else:  treshold[i, j] = 1 \* 255   tresh\_img = Image.fromarray(treshold)  self.preview\_img(tresh\_img)  except:  pass   # METHOD UNTUK MENGATUR MULTIPLY  def multiply(self,event):  try:  self.label\_parameter\_multiply.config(text=f'{(self.scale\_multiply.get()):.2f}')   img = cv.imread(str(self.file\_location))  img\_multiply = img \* self.scale\_multiply.get()  cv.imwrite('target.jpg', img\_multiply)   img = cv.imread('target.jpg')  img = cv.cvtColor(img,cv.COLOR\_BGR2RGB)  img = Image.fromarray(img)  self.preview\_img(img)  except:  pass   # METHOD UNTUK MENGATUR DIVIDE  def divide(self,event):  try:  self.label\_parameter\_divide.config(text=f'{(self.scale\_divide.get()):.2f}')   img = cv.imread(str(self.file\_location))  img\_devided = img / self.scale\_divide.get()  cv.imwrite('target.jpg', img\_devided)   img = cv.imread('target.jpg')  img = cv.cvtColor(img, cv.COLOR\_BGR2RGB)  img = Image.fromarray(img)  self.preview\_img(img)  except:  pass  # MEMBUAT SEBUAH CLASS UNTUK WINDOW ATAU MENU 2 class Window\_2: # MEMBUAT CONSTRUCTOR  def \_\_init\_\_(self,window):  # MENGATUR UKURAN ATAU DIMENSI WINDOW  window.destroy()  self.window\_2 = Tk()  self.window\_2.title("Image Browse App - 5200411434")  self.window\_2.geometry("1295x650+25+25")  style = Style(theme='darkly')   # MEMBUAT LEFT ,MIDDLE DAN RIGHT FRAME  self.left\_frame = Frame(self.window\_2, width=500, height=700, bg='#323232')  self.left\_frame.grid(row=0, column=0, padx=15, pady=5)   self.middle\_frame = Frame(self.window\_2, width=700, height=550, bg='#323232')  self.middle\_frame.grid(row=0, column=1, padx=15, pady=15)   self.right\_frame = Frame(self.window\_2,width=525, height=550,bg='#323232')  self.right\_frame.grid(row=0,column=2, padx=15, pady=15)   # MEMBUAT TOOLBARS  self.tool\_bar\_2 = Frame(self.left\_frame, width=180, height=200, bg="#3f3f3f")  self.tool\_bar\_2.grid(row=1, column=0, padx=10, pady=50, sticky='w')   self.tool\_bar\_3 = Frame(self.left\_frame, width=180, height=700, bg="#323232")  self.tool\_bar\_3.grid(row=2, column=0, padx=17, pady=20 ,sticky='s')   # MEMBUAT BUTTON UNTUK KEMBALI KE WINDOWS SEBELUMNYA  self.menu = Button(self.left\_frame, text="MENU 1", width=20, height=2, bd=0, bg="#ff7043", relief="solid",command=lambda:Window\_1(self.window\_2))  self.menu.grid(row=0, column=0, pady=15, padx=15, columnspan=2)  # MEMBUAT RB UNTUK MELAKUKAN FILTER YAITU PENJUMLAHAN ATAU PENGURANGAN CITRA  self.icon\_filter = PhotoImage(file="resources/icon/filter\_icon.png")  self.filter = Label(self.tool\_bar\_2, image=self.icon\_filter, text="Filter", compound='left', bg="#FD7014")  self.filter.grid(row=0, column=0, ipadx=55, columnspan=2)  # PENJUMLAHAN  ttk.Style().configure('info.TRadiobutton', background="#3f3f3f", foreground='white', font=('Helvetica', 8))  self.btn\_to\_gray = ttk.Radiobutton(self.tool\_bar\_2, text="Penjumlahan", value=1, style='info.TRadiobutton',command=self.penjumlahan)  self.btn\_to\_gray.grid(row=1, column=0, padx=25, pady=10, sticky='w')  # PENGURANGAN  self.btn\_to\_blue = ttk.Radiobutton(self.tool\_bar\_2, text="Pengurangan", value=2, style='info.TRadiobutton',command=self.pengurangan)  self.btn\_to\_blue.grid(row=1, column=1, padx=25, pady=10, sticky='w')  # MEMBUAT BUTTON EXIT  self.btn\_exit = Button(self.tool\_bar\_3, text="Exit", width=20, height=2, bd=0, bg="#da4453", relief="solid",command=lambda: exit())  self.btn\_exit.grid(row=4, column=0, pady=10, padx=15 ,sticky='s')   # MEMBUAT INPUTAN UNTUK GAMBAR   # MEMBACA ALAMAT GAMBAR  img\_ori = Image.open("resources/noimage.jpg")   # MENGECILKANYA DENGAN UKURAN 275,275  image\_raw = img\_ori.resize((275, 275), Image.ANTIALIAS)  self.image\_raw = ImageTk.PhotoImage(image\_raw)   # MEMBESARKAN DENGAN UKURAN 525,500  image\_result = img\_ori.resize((525, 500), Image.ANTIALIAS)  self.image\_result = ImageTk.PhotoImage(image\_result)   # MEMBUAT LABEL UNTUK GAMBAR PERTAMA  self.gambar\_1 = Label(self.middle\_frame,image=self.image\_raw)  self.gambar\_1.grid(row=0,column=0,pady=10)  self.gambar\_1.bind('<Double 1>', lambda event:self.show\_image(self.gambar\_1))   # MEMBUAT LABEL UNTUK GAMBAR KEDUA  self.gambar\_2 = Label(self.middle\_frame, image=self.image\_raw)  self.gambar\_2.grid(row=1, column=0,pady=10)  self.gambar\_2.bind('<Double 1>', lambda event:self.show\_image(self.gambar\_2))   # MEMBUAT LABEL UNTUK GAMBAR HASIL OPERASI  self.gambar\_3 = Label(self.right\_frame,image=self.image\_result)  self.gambar\_3.grid(row=0, column=0, pady=10)   # MEMBUAT METHOD UNTUK MENGAMBIL GAMBAR DARI STORAGE DAN MENAMPILKANYA  def show\_image(self,urutan):  # JIKA GAMBAR PERTAMA DI DOUBLE CLICK  if urutan == self.gambar\_1:  self.fln\_1 = filedialog.askopenfilename(initialdir=os.getcwd(), title="Select Image File",  filetypes=(("PNG Files", "\*.png"), ("JPG File", "\*.jpg"),  ("All Files", "\*.\*")))  image = Image.open(self.fln\_1)   # JIKA GAMBAR KEDUA DI DOUBLE CLICK  elif urutan == self.gambar\_2:  self.fln\_2 = filedialog.askopenfilename(initialdir=os.getcwd(), title="Select Image File",  filetypes=(("PNG Files", "\*.png"), ("JPG File", "\*.jpg"),  ("All Files", "\*.\*")))  image = Image.open(self.fln\_2)   # MENGECILKAN GAMBAR DAN MENAMPILKANYA  image\_l = image.resize((275, 275), Image.ANTIALIAS)  image\_l = ImageTk.PhotoImage(image\_l)  urutan.config(image=image\_l)  urutan.show = image\_l   # MEMBUAT METHOD UNTUK MELAKUKAN PENJUMLAHAN CITRA  def penjumlahan(self):  # MEMINDAI LOKASI GAMBAR 1 DAN MENYESUAIKAN DIMENSI GAMBARNYA  img\_1 = cv.imread(self.fln\_1)  img\_1 = cv.resize(img\_1, (275,275), interpolation=cv.INTER\_AREA)  # MEMINDAI LOKASI GAMBAR 2 DAN MENYESUAIKAN DIMENSI GAMBARNYA  img\_2 = cv.imread(self.fln\_2)  img\_2 = cv.resize(img\_2, (275, 275), interpolation=cv.INTER\_AREA)  # MENJUMLAHKAN NILAI PIXEL ATAU MATRIK YANG ADA PADA GAMBAR 1 DAN GAMBAR 2  new\_img = img\_1 + img\_2  new\_img = Image.fromarray(new\_img)  self.preview(new\_img)   # MEMBUAT METHOD UNTUK MELAKUKAN PENGURANGAN CITRA  def pengurangan(self):  # MEMINDAI LOKASI GAMBAR 1 DAN MENYESUAIKAN DIMENSI GAMBARNYA  img\_1 = cv.imread(self.fln\_1)  img\_1 = cv.resize(img\_1, (300, 300), interpolation=cv.INTER\_AREA)  # MEMINDAI LOKASI GAMBAR 2 DAN MENYESUAIKAN DIMENSI GAMBARNYA  img\_2 = cv.imread(self.fln\_2)  img\_2 = cv.resize(img\_2, (300, 300), interpolation=cv.INTER\_AREA)  # MENJUMLAHKAN NILAI PIXEL ATAU MATRIK YANG ADA PADA GAMBAR 1 DAN GAMBAR 2  new\_img = img\_1 - img\_2  new\_img = Image.fromarray(new\_img)  self.preview(new\_img)   # MENAMPILKAN GAMBAR HASIL PENJUMLAHAN / PENGURANGAN CITRA  def preview(self,image):  image\_new = image.resize((525, 500), Image.ANTIALIAS)  image\_new = ImageTk.PhotoImage(image\_new)   self.gambar\_3.config(image=image\_new)  self.gambar\_3.image = image\_new   # MEMBUAT SEBUAH OBJECT MAIN WINDOW main\_window = Window\_1() main\_window.window.mainloop()  # MENGHAPUS FILE GAMBAR HASIL DARI PENJUMLAHAN DAN PENGURANGAN CITRA JIKA ADA try:  os.remove("target.jpg") except:  exit() | | |
| **Hasil Running Aplikasi** | | |
| 1. Operasi Negative citra      1. OperasiPenjumlahan dan pengurangan Citra dengan Citra     Penjumlahan citra  Pengurangan citra   1. Operasi Perkalian dan pembagian citra dengan skalar   Pembagian dengan skalar   |  |  | | --- | --- | |  |  |   Perkalian dengan skalar   |  |  | | --- | --- | |  |  | | | |

|  |
| --- |
| **Soal 2.** |
| Berdasarkan sintaks demo di kelas berikut:   |  | | --- | | import cv2  # Reading image file  img = cv2.imread(‘./photos/data/mountain.jpg’)  cv2.imshow(‘ori.jpg’, img)  cv2.waitKey(0)  # Applying NumPy scalar multiplication on image  fimg = cv2.divide(img, 1.5)  # Saving the output image  cv2.imwrite(‘darkerLib.jpg’, fimg)  img\_result = cv2.imread(‘darkerLib.jpg’)  cv2.imshow(‘result.jpg’, img\_result)  cv2.waitKey(0)  cv2.destroyAllWindows() |   Jelaskan mengapa sintaks di atas menghasilkan gambar kebiruan seperti berikut? |
| **Penjelasan** |
| **//tuliskan penjelasan lengkap Anda di sini (disertai bukti matriks lebih baik)**  Cara kerja dari function devide diatas adalah membagi nilai matriks secara independent   |  |  | | --- | --- | |  |  |   Hal ini juga berlaku pada function yang mirip yaitu muliply sehingga walaupun bentuknya matriks tetapi tidak menggunakan kaidah perkalian matriks sebagaimana mestinya. Matriks A(0,0) dibagi B(0,0) ,A(1,0) dibagi B(1,0) ,A(0,1) dibagi B(0,1) ,dan A(1,1) dibagi B(1,1). Sehingga nilai matriks A pada koordinat yang sama dilakukan operasi dengan nilai Matriks B dengan koordinat yang sama pula. Jadi yang dioperasikan hanya bagian nilainya saja dengan kaidah operasi yang sama dengan perkalian atau pembagian pada umumnya. Hal tersebut jugalah yang menyebabkan apabila menggunakan function ini kedua gambar harus memiliki panjang dan lebar yang sama persis. Apabila salah satu tidak sama maka akan error tetapi beda halnya jika yang dikalikan adalah sebuah skalar. Contohnya adalah seperti berikut.   |  |  | | --- | --- | | (a) |  |   Terdapat 2 buah gambar berukuran 3x3 dan masing-masing memiliki nilai pixelnya sendiri dan kedua gambar tersebut merupakan gambar dengan true color yaitu gambar yang memiliki 3 channel warna (red, green ,blue). Akan diberikan operasi pembagian / devide menggunakan function yang disediakan oleh opencv sebelumnya dengan syarat kedua gambar harus memiliki ukuran yang sama. Sehingga setelah dilakukan operasi devide maka akan menghasilkan gambar beserta nilai pixelnya seperti yang ada pada gambar dibawah.   |  |  | | --- | --- | |  |  |      |  |  | | --- | --- | |  |  |   Nilai pixel setelah dioperasikan berubah menjadi lebih kecil dari sebelumnya bahkan ada yang sampai 0 .Hal ini karena gambar kiri dibagi dengan kanan sebagai contoh 151 dibagi dengan 38 adalah 3,97368 akan tetapi oleh opencv dibulatkan keatas menjadi 4. Begitupun seterusnya untuk semua nilai pixel yang ada akan dioperasikan sesuai dengan nilai koordinat yang sama. Gambar hasil operasi menjadi berwarna hitam karena nilainya hampir semua mendekati 0 dan 0 sendiri merupakan kode untuk warna hitam, dengan begitu maka gambar tampak berwarna hitam. Contoh diatas merupakan operasi matriks dengan matriks dan akan dilanjut dengan operasi matiks x scalar ataupun kebalikanya scalar x matriks. Prinsipnya tetap sama akan tetapi terdapat sedikit perbedaan karena dioperasi dengan sebuah skalar yang mana skalar hanya memiliki ukuran 1x1.   |  |  | | --- | --- | |  |  |   Gambar diatas merupakan hasil pembagian matriks dengan skalar 1,5 hasil sebenarnya adalah 100,6 yang kemudian dibulatkan menjadi 101 tetapi terjadi perbedaan disini nilai pixel selanjutnya harusnya bernilai 122 akan tetapi oleh opencv ditulis 0. Hal ini dapat terjadi karena ukuran kedalaman channelnya anatara skalar dengan gambar berbeda. Gambar memiliki kedalam 3 channel rgb sedangkan skalar hanya ada 1 saja sehingga mengakibatkan nilai pixel pada index awal saja yang dikali dengan skalar sedangkan 2 nilai lainya akan dikali dengan 0. Dan dikarenakan format di opencv adalah bgr bukan rgb maka yang berada di index awal adalah channel blue sehingga hanya blue saja yang dioperasikan dengan skalar dan karena hal itulah gambar menjadi warna biru.  Kemudian pada soal hasil gelap kebiruan diatas terjadi karena hasil perkalian matriks (gambar) dengan skalar dimana matriksnya memiliki 3 channel (rgb) sedangkan skalar hanya satu sehingga mengakibatkan super position. Yaitu hanya channel yang urutannya di depan saja yang mengalami operasi dengan skalar sedangkan yang 2 sisanya akan dikali dengan 0. Disini warna biru terjadi karena library opencv membaca format warna secara BGR bukan RGB sehingga warna biru yang berada paling depan akan dioperasikan dengan skalar. Jadi apabila nilai depannya berupa channel merah maka gambar akan menjadi kemerah-merahan. Dan warna yang gelap dikarenakan hasil pembagian dengan skalar sehingga nilai pixelnya menjadi lebih kecil. Oleh karena itu untuk mengurandi tingkat kecerahan gambar dilakukan dengan cara membagi nilai pixel sedangkan untuk meningkatkan kecerahan dilakukan dengan cara mengkali nilai pixel. |