|  |  |  |
| --- | --- | --- |
|  | **Lembar Kerja Mahasiswa**  **Mata Kuliah Pengolahan Citra Digital Praktik (203311-20)**  **Program Studi Informatika**  **Fakultas Sains & Teknologi – Universitas Teknologi Yogyakarta** | |
|  | **Identitas Mahasiswa** | |
| **Nama** | **Arieska Restu Harpian Dwika** |
| **NPM** | **5200411488** |
| **Kelompok Prak** | **Kel. I** |
|  | | |
| **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**  # 5200411488 - Arieska Restu Harpian Dwika  import cv2  import numpy as np  import matplotlib.pyplot as plt  import matplotlib.image as mpimg  from tkinter import \*  from tkinter import filedialog  from ttkbootstrap import Style  from tkinter import ttk  import tkinter as tk  import os  from PIL import Image, ImageTk  def setOriginal1(*img*):      imgTk = ImageTk.PhotoImage(*img*)      lblOriImg1.configure(*image*=imgTk)      lblOriImg1.image = imgTk      lblOriImg1.pack()  def setOriginal2(*img*):      imgTk = ImageTk.PhotoImage(*img*)      lblOriImg2.configure(*image*=imgTk)      lblOriImg2.image = imgTk      lblOriImg2.pack()  def setResult(*img*):      imgTk = ImageTk.PhotoImage(*img*)      lblResultImg.configure(*image*=imgTk)      lblResultImg.image = imgTk      lblResultImg.pack()  def opencv2Pill(*img*):  *img* = cv2.cvtColor(*img*, cv2.COLOR\_BGR2RGB)      imgPill = Image.fromarray(*img*)      return imgPill    def resizeImg(*img*):      width, height = 320, 240  *img* = cv2.resize(*img*, (width, height))      return *img*  def clipping(*intensity*):      if *intensity* < 0:          return 0      if *intensity* > 255:          return 255      return *intensity*  def browseImage1():      global fln1      fln1 = filedialog.askopenfilename(*initialdir*=os.getcwd(), *title*="Select Image File",  *filetypes*=(                                          ("All Files", "\*.\*",),                                          ("PNG File", "\*.png"),                                          ("JPG File", "\*.jpg"))                                      )      img = opencv2Pill(resizeImg(cv2.imread(fln1)))      setOriginal1(img)  def browseImage2():      global fln2      fln2 = filedialog.askopenfilename(*initialdir*=os.getcwd(), *title*="Select Image File",  *filetypes*=(                                          ("All Files", "\*.\*",),                                          ("PNG File", "\*.png"),                                          ("JPG File", "\*.jpg"))                                      )      img = opencv2Pill(resizeImg(cv2.imread(fln2)))      setOriginal2(img)  def adding():      global fln1, fln2      img1 = opencv2Pill(resizeImg(cv2.imread(fln1)))      px1 = img1.load()      img2 = opencv2Pill(resizeImg(cv2.imread(fln2)))      px2 = img2.load()      hor = img1.size[0]      ver = img1.size[1]      imgResult = Image.new("RGB", (hor, ver))      pxResult = imgResult.load()      for x in range(hor):          for y in range(ver):              r = clipping(px1[x, y][0] + px2[x, y][0])              g = clipping(px1[x, y][1] + px2[x, y][1])              b = clipping(px1[x, y][2] + px2[x, y][2])              pxResult[x, y] = (r, g, b)      setResult(imgResult)  def subtracting():      global fln1, fln2      img1 = opencv2Pill(resizeImg(cv2.imread(fln1)))      px1 = img1.load()      img2 = opencv2Pill(resizeImg(cv2.imread(fln2)))      px2 = img2.load()      hor = img1.size[0]      ver = img1.size[1]      imgResult = Image.new("RGB", (hor, ver))      pxResult = imgResult.load()      for x in range(hor):          for y in range(ver):              r = clipping(px1[x, y][0] - px2[x, y][0])              g = clipping(px1[x, y][1] - px2[x, y][1])              b = clipping(px1[x, y][2] - px2[x, y][2])              pxResult[x, y] = (r, g, b)      setResult(imgResult)    if \_\_name\_\_ == '\_\_main\_\_':      fln1, fln2 = None, None      style = Style()      window = style.master      frm = ttk.Frame(window, *style*='primary.TFrame')      frm.pack\_propagate(0)      frm.pack(*fill*=tk.BOTH, *expand*=1)      frmImgOri = ttk.Frame(frm, *style*='secondary.TFrame', *width*=900, *height*=500)      frmImgOri.grid(*row*=0, *column*=0, *padx*=25, *pady*=25)      frmImgOri1 = ttk.Frame(frmImgOri, *style*='info.TFrame', *width*=320, *height*=240)      frmImgOri1.pack\_propagate(0)      frmImgOri1.pack(*side*="left", *padx*=20, *pady*=30)      frmBtn = ttk.Frame(frmImgOri, *style*='secondary.TFrame', *width*=100, *height*=200)      frmBtn.pack(*side*="left", *padx*=20, *pady*=30)      frmImgOri2 = ttk.Frame(frmImgOri, *style*='info.TFrame', *width*=320, *height*=240)      frmImgOri2.pack\_propagate(0)      frmImgOri2.pack(*side*="left", *padx*=20, *pady*=30)      frmImgRes = ttk.Frame(frm, *style*='secondary.TFrame', *width*=320, *height*=240)      frmImgRes.pack\_propagate(0)      frmImgRes.grid(*row*=1, *column*=0, *padx*=15, *pady*=30)      frmImgResult = ttk.Frame(frmImgRes, *style*='info.TFrame', *width*=320, *height*=240)      frmImgResult.pack\_propagate(0)      frmImgResult.grid(*row*=1, *column*=0, *padx*=30, *pady*=30)      btnBrowse1 = ttk.Button(frmBtn, *text*='Browse Image 1', *style*='info.TButton', *cursor*="hand2", *width*=14, *command*=browseImage1)      btnBrowse1.pack(*side*='top', *pady*=10)      btnBrowse2 = ttk.Button(frmBtn, *text*='Browse Image 2', *style*='info.TButton', *cursor*="hand2", *width*=14, *command*=browseImage2)      btnBrowse2.pack(*side*='top', *pady*=10)      btnAdding = ttk.Button(frmBtn, *text*='+', *style*='success.TButton', *cursor*="hand2", *width*=2, *command*=adding)      btnAdding.pack(*side*='top', *pady*=10)      btnSubtract = ttk.Button(frmBtn, *text*='-', *style*='success.TButton', *cursor*="hand2", *width*=2, *command*=subtracting)      btnSubtract.pack(*side*='top', *pady*=10)      btnExit = ttk.Button(frmBtn, *text*='Exit', *style*='danger.TButton', *cursor*="hand2", *command*=lambda: exit())      btnExit.pack(*side*='top', *pady*=10)      lblOriImg1 = ttk.Label(frmImgOri1)      lblOriImg2 = ttk.Label(frmImgOri2)      lblResultImg = ttk.Label(frmImgResult)      window.title("Adding or Subtracting Images  - 5200411488")      # window.geometry("1280x720")      # window.resizable(0, 0)      window.mainloop() | | |
| **Hasil Running Aplikasi** | | |
| **//paste-kan tampilan aplikasi Anda di sini**    Gambar 1 Penjumlahan dua buah citra    Gambar 2 Pengurangan dua buah citra | | |

|  |
| --- |
| **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)**  Sintaks tersebut dapat menghasilkan gambar kebiruan karena operasi yang dilakukan di fungsi cv2.divide() pada baris ke-9. Fungsi cv2.divide() melakukan pembagian per elemen dari dua buah citra atau citra dengan skalar. Citra yang dimasukkan ke dalam fungsi cv2.divide(), akan diambil channel warna biru dan dikalikan dengan skalar yang dimasukkan ke parameter fungsi tersebut. Sedangkan untuk channel warna lainnya akan bernilai nol. |