

**LAPORAN TUGAS 2**  
**16TIN5033 PENGOLAHAN CITRA DIGITAL**



Disusun Oleh:  
Muhammad Aziz Taufiqurrahman    201524014

**PROGRAM STUDI D4 TEKNIK INFORMATIKA**  
**JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA**  
**POLITEKNIK NEGERI BANDUNG**  
**2022**

## KODE PROGRAM

Kode program saya buat menggunakan teknologi PySimpleGUI untuk membuat tampilannya / GUI. Kemudian menggunakan numpy, cv2 dan bahasa pemrograman python. IDE yang digunakan adalah Visual Studio Code. Berikut merupakan source code dari program :

```
from tokenize import Triple
import PySimpleGUI as sg
import cv2 as cv
import numpy as np

def encode_img(img):
    return cv.imencode(".png", img)[1].tobytes()

def main():
    sg.theme("Black")

    layout = [
        [sg.Image(filename="", key="image")],
        [sg.Button("Load", size=(10, 1)),
         sg.Button("Restore", key="restore-btn", size=(10, 1), visible=False)],
        [sg.Text("Red", key="text-r", visible=False), sg.Slider(range=(0, 10),
default_value=0,
                                                                    visible=False,
orientation="h", key="slider-r", enable_events=True)],
        [sg.Text("Green", key="text-g", visible=False), sg.Slider(range=(0, 10),
default_value=0,
                                                                    visible=False,
orientation="h", key="slider-g", enable_events=True)],
        [sg.Text("Blue", key="text-b", visible=False), sg.Slider(range=(0, 10),
default_value=0,
                                                                    visible=False,
orientation="h", key="slider-b", enable_events=True)]
    ]

    window = sg.Window("Week 2 PCD", layout)

    image_element = window["image"]

    img = None
    curr_slider_value = [0, 0, 0]
    while True:
```

```

event, values = window.read()

if event == "Load":
    filename = sg.popup_get_file("File Gambar", file_types=(
        ("Images", "*.JPG;*.JPEG;*.JPE;*.JP2;*.BMP;*.DIB"), ))

    if filename != None and filename != "":
        curr_slider_value = [0, 0, 0]
        img = cv.imread(filename, cv.IMREAD_COLOR)

        blue=img[:, :, 0]
        green=img[:, :, 1]
        red=img[:, :, 2]

        bl=np.average(blue)
        gr=np.average(green)
        rd=np.average(red)

        print = sg.Print
        print('Warna Biru : ', bl)
        print('Warna Hijau : ', gr)
        print('Warna Merah : ', rd)

        if bl>gr:
            if bl>=rd:
                result=print('Warna Dominan: BLUE')
                print(bl)
            else:
                if gr>=rd:
                    result=print('Warna Dominan: GREEN')
                    print(gr)
                else:
                    result=print('Warna Dominan: RED')
                    print(rd)

        window["slider-r"].update(visible=True)
        window["slider-g"].update(visible=True)
        window["slider-b"].update(visible=True)

        window["text-r"].update(visible=True)
        window["text-g"].update(visible=True)
        window["text-b"].update(visible=True)

        window["restore-btn"].update(visible=True)

```

```

        image_element.update(data=encode_img(img))

    if event == "restore-btn":
        image_element.update(data=encode_img(img))

        window["slider-r"].update(0)
        window["slider-g"].update(0)
        window["slider-b"].update(0)

        if int(values["slider-r"]) != curr_slider_value[2] or int(values["slider-
g"]) != curr_slider_value[1] or int(values["slider-b"]) != curr_slider_value[0]:
            curr_slider_value[2] = int(values["slider-r"])
            curr_slider_value[1] = int(values["slider-g"])
            curr_slider_value[0] = int(values["slider-b"])

        if not img is None:
            temp_img = np.copy(img)
            temp_img[:, :, 2] += curr_slider_value[2]
            temp_img[:, :, 1] += curr_slider_value[1]
            temp_img[:, :, 0] += curr_slider_value[0]

            image_element.update(data=encode_img(temp_img))

    if event in ("Quit", sg.WIN_CLOSED):
        break

window.close()

main()

```

## 1. Grayscale

Grayscale : suatu istilah untuk menyebutkan satu citra yang memiliki warna abu-abu, hitam, dan putih. Grayscale adalah koleksi atau kisaran corak monokromik (abu-abu), mulai dari putih murni di ujung yang paling terang hingga hitam murni di ujung yang berlawanan.

Grayscale hanya berisi informasi pencahayaan (kecerahan) dan tidak ada informasi warna; itulah sebabnya pencahayaan maksimum putih dan pencahayaan hitam nol; segala sesuatu di antaranya adalah warna abu-abu. Itulah sebabnya gambar grayscale hanya berisi nuansa abu-abu dan tidak ada warna. Grayscale juga dikenal sebagai akromatik. Sejak gambar digital ditampilkan menggunakan kombinasi merah, hijau, dan biru (RGB), setiap pixel

memiliki tiga nilai luminance terpisah. Oleh karena itu, tiga nilai-nilai ini harus digabungkan menjadi satu nilai saat melepas warna dari sebuah gambar.

Citra grayscale merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pikselnya, artinya nilai dari Red = Green = Blue. Nilai-nilai tersebut digunakan untuk menunjukkan intensitas warna. Citra yang ditampilkan dari citra jenis ini terdiri atas warna abu-abu, bervariasi pada warna hitam pada bagian yang intensitas terlemah dan warna putih pada intensitas terkuat. Citra grayscale berbeda dengan citra "hitam-putih", dimana pada konteks computer, citra hitam putih hanya terdiri atas 2 warna saja yaitu "hitam" dan "putih" saja. Pada citra grayscale warna bervariasi antara hitam dan putih, tetapi variasi warna diantaranya sangat banyak. Citra grayscale seringkali merupakan perhitungan dari intensitas cahaya pada setiap piksel pada spektrum elektromagnetik single band. Untuk mengubah citra berwarna yang mempunyai nilai matrik masing-masing R, G dan B menjadi citra grayscale dengan nilai X, maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai R, G dan B

- GrayScale dengan OpenCV

```
import cv2

image = cv2.imread('FileName')
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

cv2.imshow('Original image',image)
cv2.imshow('Gray image', gray)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

- Grayscale tanpa OpenCV

```
import numpy as np

def rgb2gray(rgb):

    r, g, b = rgb[:, :, 0], rgb[:, :, 1], rgb[:, :, 2]
    gray = 0.2989 * r + 0.5870 * g + 0.1140 * b

    return gray
```

## **2. Biner**

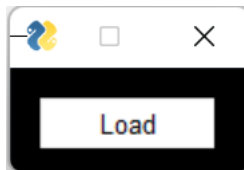
Citra biner (binary image) adalah citra digital yang hanya memiliki 2 kemungkinan warna, yaitu hitam dan putih. Citra biner disebut juga dengan citra W&B (White&Black) atau citra monokrom. Hanya dibutuhkan 1 bit untuk mewakili nilai setiap piksel dari citra biner. Pembentukan citra biner memerlukan nilai batas keabuan yang akan digunakan sebagai nilai patokan. Piksel dengan derajat keabuan lebih besar dari nilai batas akan diberi nilai 1 dan sebaliknya piksel dengan derajat keabuan lebih kecil dari nilai batas akan diberi nilai 0. Citra biner sering sekali muncul sebagai hasil dari proses pengolahan, seperti segmentasi, pengambangan, morfologi ataupun dithering. Fungsi dari binerisasi sendiri adalah untuk mempermudah proses pengenalan pola, karena pola akan lebih mudah terdeteksi pada citra yang mengandung lebih sedikit warna

## **3. Inverse**

Invers adalah salah satu metode pengolahan citra. Proses invert membuat citra gambar menjadi tampak negative pada susunan warna RGB, namun berbeda dengan proses gambar negative. Invers dihasilkan dari nilai maximum dari RGB (255,255,255) dikurang nilai intensitas warna RGB pada gambar.

## TAMPILAN ANTARMUKA PROGRAM

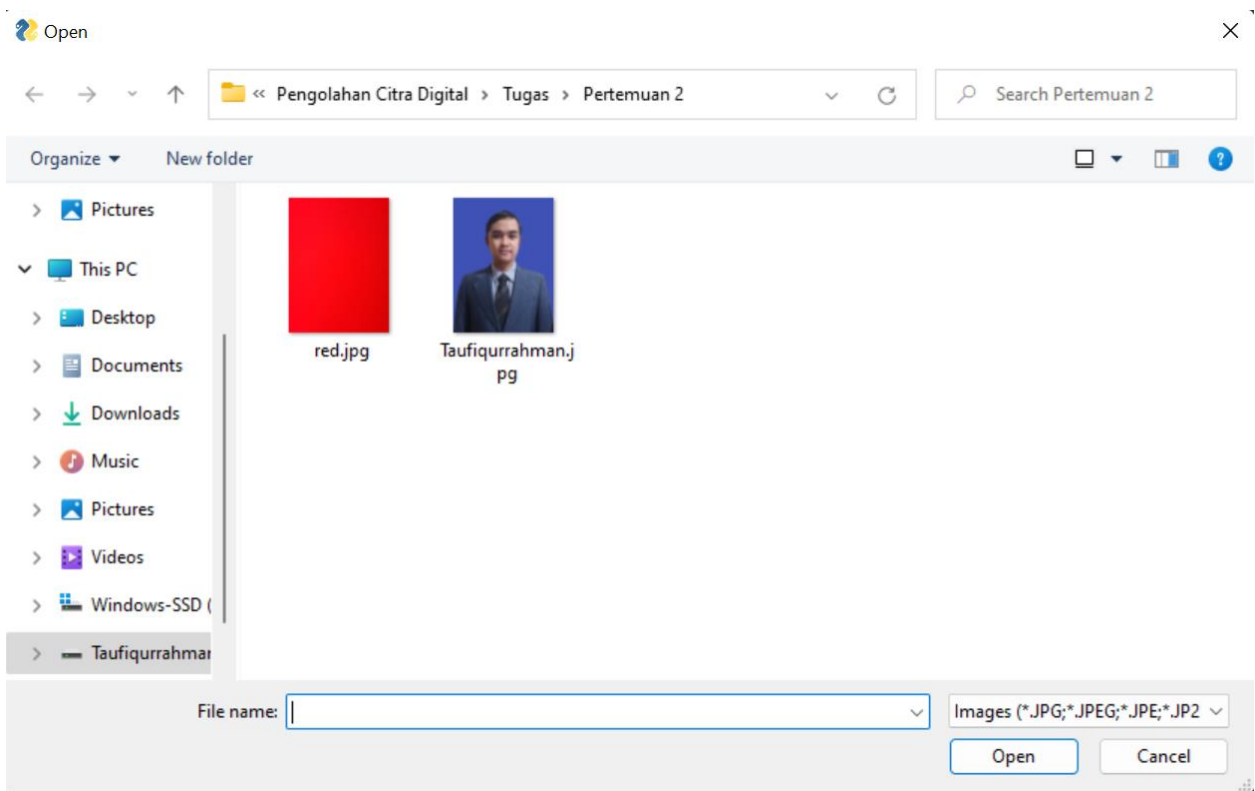
### 1. Tampilan untuk opsi *Load*



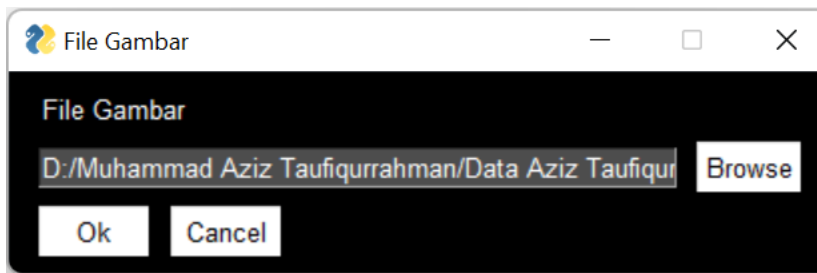
### 2. Tampilan saat diminta untuk *input image*



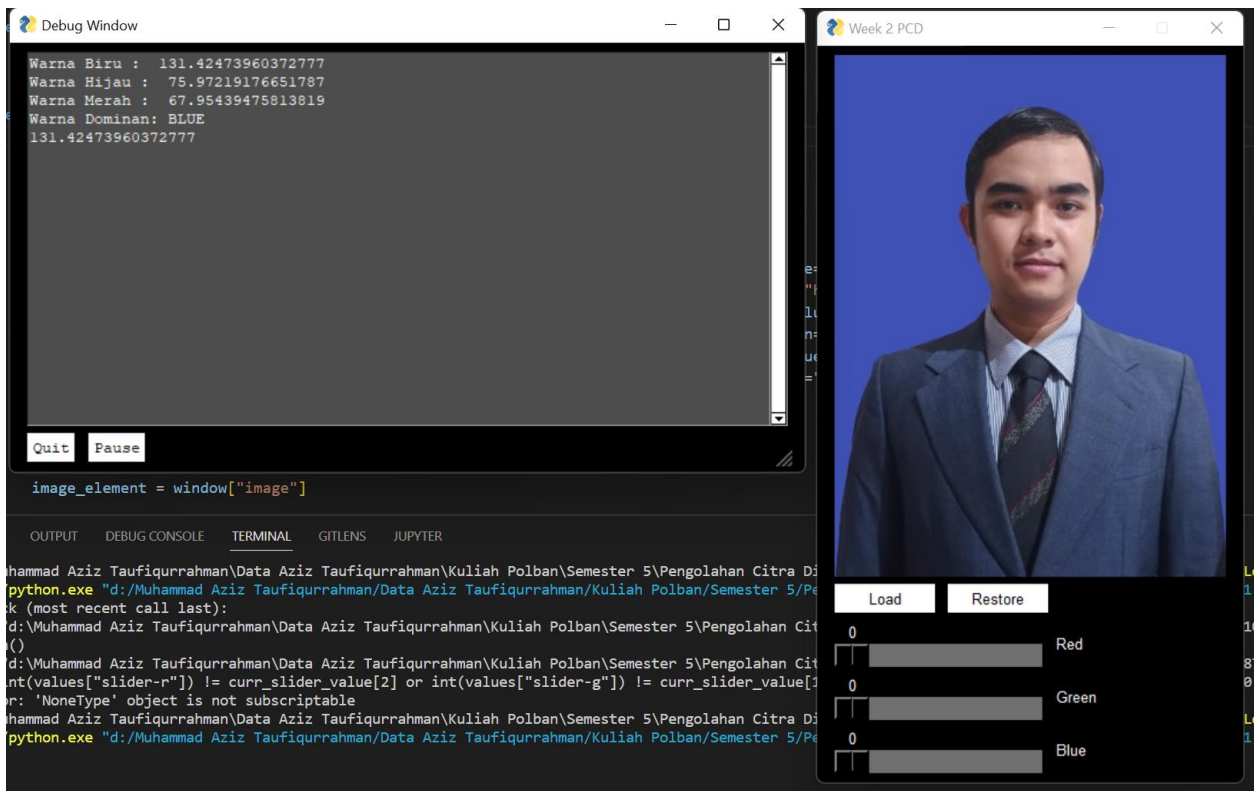
### 3. Tampilan saat memilih file image



#### 4. Tampilan saat file sudah dipilih



#### 5. Tampilan hasil dari program beserta dengan pilihan dan hasil dari RGB image

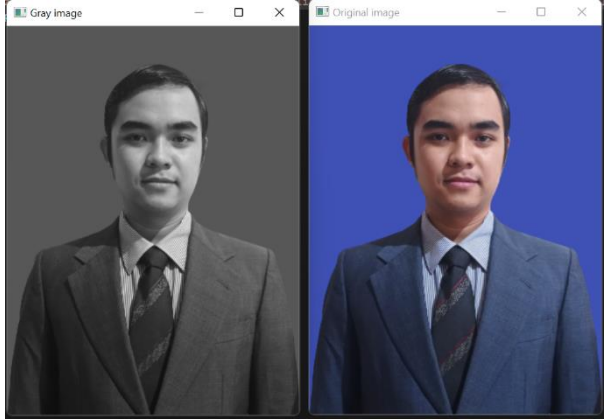


Link GitHub : <https://github.com/aziztaufiqurrahman/PCD.git>



## HASIL EKSEKUSI PROGRAM

### Grayscale

Masukan	Keluaran
<pre>import cv2  image = cv2.imread('D:\Muhammad Aziz Taufiqurrahman\Data Aziz Taufiqurrahman\Kuliah Polban\Semester 5\Pengolahan Citra Digital\Tugas\Pertemuan 2\Taufiqurrahman.png')  gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)  cv2.imshow('Original image',image)  cv2.imshow('Gray image', gray)  cv2.waitKey(0)  cv2.destroyAllWindows()</pre>	 The output shows two windows side-by-side. The left window, titled 'Gray image', displays a grayscale version of a man in a suit. The right window, titled 'Original image', displays the same man in a suit with a blue background.

### Arithmetic Operations

Masukan 1	Masukan 2	Keluaran

## KENDALA

Pada pengerjaan tugas ini kendala yang saya hadapi adalah saat pembuatan GUI menggunakan bahasa pemrograman Python. Kemudian yang saya alami juga cara input image menggunakan button dari computer agar bisa ditampilkan pada GUI nya. Dan yang paling penting itu sulitnya menampilkan hasil dari RGB suatu image ke GUI, dikarenakan hasil RGB nya ditampilkan terus menerus di dalam terminal.

## SOLUSI

Solusi yang saya lakukan adalah dengan menginstall dan menggunakan teknologi dari PySimpleGUI untuk Python. Hal tersebut ditujukan agar mempermudah proses pengerjaan GUI dari aplikasi. Kemudian solusi untuk input image, saya mendapatkan jawaban dari beberapa sumber rujukan seperti *stackoverflow*, *github*, *geeksforgeeks*, dll. Untuk solusi permasalahan menampilkan hasil RGB adalah dengan menggunakan code *print = sg.Print* sehingga yang asalnya ditampilkan di dalam terminal, bisa ditampilkan di dalam GUI.

## LESSON LEARN

Banyak sekali pelajaran yang saya dapatkan dari praktikum kali ini, seperti proses pembuatan GUI menggunakan bahasa Python kemudian proses mengambil data dari computer agar bisa ditampilkan pada GUI aplikasi, bisa mendeteksi RGB suatu image kemudian menampilkan hasilnya, serta bisa mengatur kuantitas RGB dari suatu image. Intinya dengan praktikum kali ini saya bisa eksplorasi banyak sekali hal yang sebelumnya tidak saya ketahui dan saya sangat *excited* dengan pengalaman praktikum kali ini disebabkan bisa mendapatkan banyak sekali *insight*.

## REFERENSI :

[https://docs.opencv.org/4.x/d0/d86/tutorial\\_py\\_image\\_arithmetics.html](https://docs.opencv.org/4.x/d0/d86/tutorial_py_image_arithmetics.html)

[https://docs.opencv.org/4.x/d7/d4d/tutorial\\_py\\_thresholding.html](https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html)

[https://docs.opencv.org/3.4/de/d25/imgproc\\_color\\_conversions.html](https://docs.opencv.org/3.4/de/d25/imgproc_color_conversions.html)

<https://www.geeksforgeeks.org/arithmetic-operations-on-images-using-opencv-set-1-addition-and-subtraction/>

<https://stackoverflow.com/questions/66381996/print-output-in-the-gui-screen>

<https://pengertian.apa-itu.net/apa-yang-dimaksud-dengan-grayscale.html>

<https://ilmuelektro.id/grayscale-adalah/>

<https://techtutorialsx.com/2018/06/02/python-opencv-converting-an-image-to-gray-scale/>

<https://stackoverflow.com/questions/67514577/convert-color-image-to-grayscale-in-python-without-opencv>

<https://repository.usm.ac.id/files/skripsi/C41A/2014/C.431.14.0018/C.431.14.0018-05-BAB-II-20190226114654.pdf>

<https://pdfcoffee.com/implementasi-negative-image-invers-dan-brightness-pdf-free.html>

<https://www.hackerearth.com/practice/notes/extracting-pixel-values-of-an-image-in-python/>

<https://www.geeksforgeeks.org/python-pil-getpixel-method/>

<https://www.adamsmith.haus/python/answers/how-to-find-the-rgb-value-of-a-pixel-in-python>