

MODUL 2 – Penggunaan Numpy dan OpenCV serta Pengenalan Aplikasi-aplikasi Pengolahan Citra

A. TUJUAN

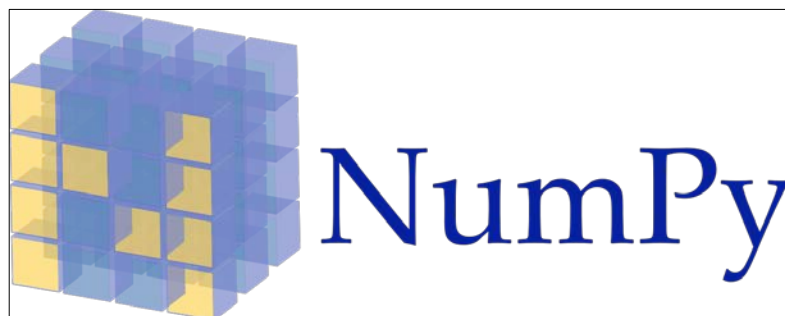
1. Mampu membuka file image dari Google Drive pribadi
2. Mampu memahami dasar-dasar library OpenCV di Python
3. Mampu memahami channel warna pada OpenCV dan konversinya

B. ALAT DAN BAHAN

1. PC/LAPTOP
2. Github
3. *Google Colaborator*

C. ULASAN TEORI

NumPy (Numerical Python) merupakan library Python untuk data scientist dengan kemampuan untuk membentuk objek N-dimensional array. NumPy merupakan bagian ekosistem Python dari tool open source SciPy. Kemampuan yang dimiliki ini memiliki kemiripan dengan pada Python. Akan tetapi konsumsi memory dan NumPy array lebih kecil dan runtime yang lebih cepat dibandingkan list pada Python. Oleh karena itu NumPy juga memudahkan dalam mengeksekusi operasi terkait Vector (1-d array) dan Matrix (2-d array).



Sebelum menggunakan NumPy terlebih dahulu harus melakukan import package Numpy menggunakan `pip install numpy`. Akan tetapi jika Numpy sudah terinstall atau kita sudah menggunakan fasilitas google Colab maka NumPy bisa langsung di import menggunakan `import numpy as np`.

Pembuatan array dengan NumPy diterapkan dengan penambahan tipe data dengan parameter **dtype**. Berikut ini adalah beberapa tipe yang terdapat pada NumPy :

bool_	Boolean (True or False) stored as a byte
int_	Default integer type
intc	Identical to C int e.g int32 in64
intp	Integer used for indexing
int8	Byte (-128 to 127)
int16	Integer (-32768 to 32767)
int32	Integer (-2147483648 to 2147483647)
int64	Integer (-9223372036854775808 to 9223372036854775807)
uint8	Unsigned integer (0 to 255)
uint16	Unsigned integer (0 to 65535)
uint32	Unsigned integer (0 to 4294967295)
uint64	Unsigned integer (0 to 18446744073709551615)
float16	Half precision float: sign bit, 5 bits exponent, 10 bits mantissa
float32	Single precision float: sign bit, 8 bits exponent, 23 bits mantissa
float64	Double precision float: sign bit, 11 bits exponent, 52 bits mantissa
complex64	Complex number, represented by two 32-bit floats (real and imaginary components)
complex128	Complex number, represented by two 64-bit floats (real and imaginary components)

(Sumber : <https://vsvaibhav2016.medium.com/basics-of-numpy-python-for-data-analysis-45b0c43f591b>)

Array NumPy disebut sebagai ndarray, atau dikenal juga dengan sebutan array. Untuk mengatasi permasalahan dengan array khususnya array multidimensi dengan NumPy, dimensi NumPy dimensi disebut sebagai axis, dan banyaknya axis disebut rank. Untuk ruang 3D, koordinat titik diruang 3D 1 axis dengan panjang 3 dapat dinyatakan seperti [2,1,1]. Sedangkan panjang axis tersebut adalah 3. Selain contoh tersebut array seperti ini [[1. , 1. , 1.], [2. , 2. , 2.]] adalah contoh array dengan rank 2 yang memiliki 2 axis dimana axis pertama panjangnya 2, dan axis kedua memiliki panjang 3. Berikut ini adalah beberapa atribut penting dari ndarray :

Atribut	Kegunaan
ndarray.ndim	Jumlah dimensi array. Pada bahasa Python biasa disebut rank.
ndarray.shape	Ukuran array di setiap dimensi. Matriks dengan n baris dan m kolom atau matriks (n,m).
ndarray.size	Jumlah elemen suatu array atau merupakan perkalian elemen dari shape
ndarray.dtype	Merupakan tipe data Python standar suatu elemen dalam array

<code>ndarray.itemsize</code>	Ukuran dalam byte dari setiap elemen array. Contoh : Sebuah array yang memiliki elemen dengan tipe float64 memiliki itemsize 8 (= 64/8)
<code>ndarray.data</code>	Buffer berisikan elemen array

Terdapat beberapa metode utama NumPy array yang digunakan untuk memanipulasi data sebagai berikut :

1. Indexing

Metode untuk mengakses suatu elemen array. Contohnya : `a[5]`. A adalah suatu array NumPy, sedangkan 5 artinya elemen ke-5 pada array tersebut.

2. Slicing

Metode untuk mengakses array khususnya subarray (beberapa elemen sekaligus). Akses subarray dilakukan dengan karakter khusus (:). Contohnya : `a[2:6]`, artinya akses elemen array dilakukan mulai dari index 2 sampai dengan 5(6-1). `a[:5]`, artinya akses 5 elemen pertama dari suatu array. `a[::2]`, artinya akses array dilakukan mulai index 0, sampai dengan index akhir, dengan jarak 2 elemen setiap aksesnya.

3. Iterating

Metode akses array secara iteratif. Contohnya :

```
for row in b:
    print (row )
```

Artinya setiap array diakses dan ditampilkan per baris

```
for element in b.flat :
    print ( element )
```

Artinya setiap array diakses dan ditampilkan per elemen di setiap baris

4. Reshaping

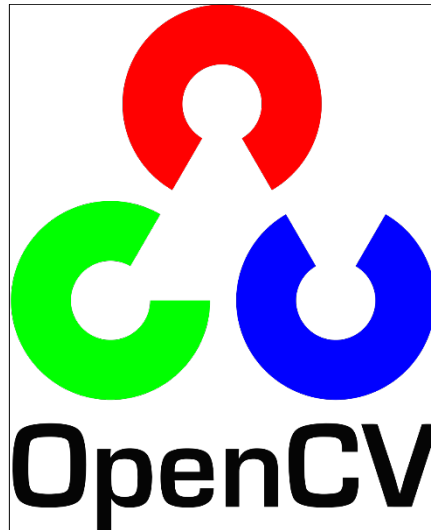
Metode untuk mengubah bentuk array. Contohnya :

```
a=np.array ([ 3. , 6. , 1. , 5. , 3. , 2. , 0. , 0. , 2. , 5. ,
1. , 4.])
a.reshape (6 ,2)
```

Artinya array a akan diubah bentuk menjadi 6 baris 2 kolom menjadi

```
array([[3., 6.],
       [1., 5.],
       [3., 2.],
       [0., 0.],
       [2., 5.],
       [1., 4.]])
```

OpenCV (Open Source Computer Vision Library) dikembangkan oleh Intel Corporation dan merupakan library open source yang disediakan untuk programming terkait citra digital dan banyak metode AI (Artificial Intelligence). OpenCV memiliki banyak fitur terkait citra digital seperti pengenalan wajah, pelacakan wajah, deteksi wajah, Kalman filtering, algoritma sederhana pada Computer Vision untuk low level API, dll. OpenCV awalnya merupakan library untuk bahasa pemrograman C/C++ dan kini dikembangkan ke bahasa lain juga seperti Python, java, dan juga matlab.



Versi pertama OpenCV dikenalkan pada tahun 1999 oleh Gary Bradsky dan mulai dan mulai dirilis pada tahun 2000. OpenCV versi pertama ini memerlukan library dari Intel Image Processing dalam penggunaannya. Akan tetapi ketergantungan akan library tersebut sudah dihilangkan dan sekarang berdiri sebagai library standalone dan multiplatform sehingga dapat digunakan pada berbagai operating system baik window, linux, macOSX maupun android.

OpenCV memiliki 5 (lima) library dasar sebagai berikut :

1. CV
Mendukung algoritma image processing dan juga computer vision
2. ML
Mendukung algoritma machine learning
3. Highgui
Mendukung GUI, image, video I/O
4. CXCore
Mendukung struktur data, XML,
5. CvAux
Khusus modul computer vision

Berdasarkan library utama yang dimiliki OpenCV terdapat beberapa fitur utama sebagai berikut :

1. Image and video I/O
2. Computer Vision secara umum dan pengolahan citra digital (untuk low dan mid level API)
3. Modul computer vision high level
4. Metode untuk AI dan machine learning
5. Sampling gambar dan transformasi
6. Metode untuk menciptakan dan menganalisa gambar biner
7. Metode untuk memperhitungkan pemodelan 3D
8. Pendeteksi gerak
9. Analisis Struktural

10. Kalibrasi Kamera
11. Image Labelling
12. DII

Aplikasi terkait pengolahan citra diterapkan dalam berbagai solusi pemecahan masalah berbagai bidang seperti berikut ini :

1. Militer

Beberapa contoh aplikasi pada bidang militer seperti teropong malam hari, identifikasi pesawat musuh dengan radar, dan sasaran peluru kendali dengan sensor virtual.

2. Kesehatan

Beberapa contoh aplikasi pada bidang kesehatan seperti deteksi patah tulang, rekontruksi foto janin, deteksi kanker, dan sebagainya.

3. Biologi

Contoh aplikasi pada bidang biologi seperti pengenalan kromosom dengan gambar mikroskopis.

4. Pendidikan

Contoh aplikasi pada bidang pendidikan seperti pengolahan pendaftaran mahasiswa dengan scanner.

5. Remote Sensing

Beberapa contoh aplikasi pada bidang kesehatan seperti pemetaan batas wilayah daerah, pengenalan jenis lapisan batuan bawah permukaan bumi

6. Kepolisian/Hukum

Beberapa contoh aplikasi pada bidang kesehatan seperti pengenalan pola sidik jari, rekonstruksi wajah pelaku kejahatan,

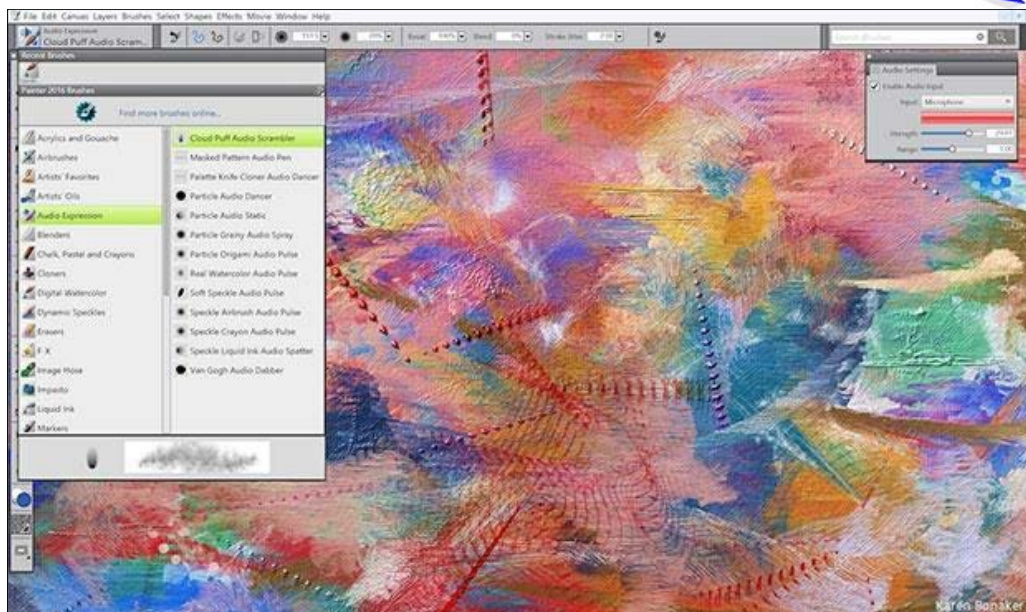
7. Perdagangan

Beberapa contoh aplikasi pada bidang perdagangan seperti pembaca barcode di swalayan, pengenalan huruf atau angka otomatis

Terdapat beberapa pengolah citra digital yang dapat menghasilkan berbagai aplikasi yang bermanfaat bagi seperti yang sudah dijelaskan sebelumnya sebagai berikut :

1. Corel Painter

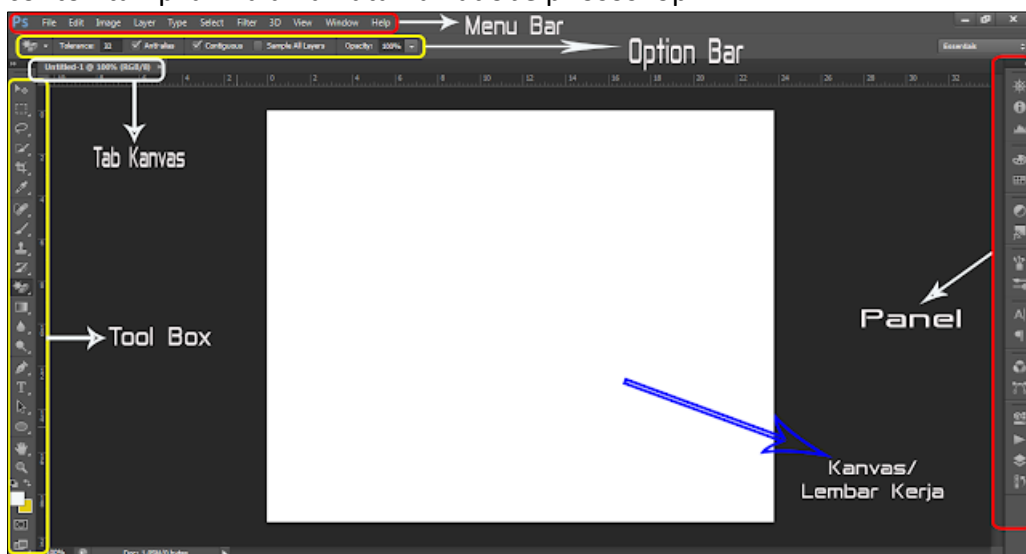
Corel painter merupakan tool pengolah citra digital yang cukup unik dengan keutamaannya yang memiliki 131 jenis brush dengan teknologi canggih seperti Dynamic Speckles (fitur yang menggabungkan control ketebalan brush dengan hukum fisika partikel sehingga brush menjadi lembut dan dinamis), selain itu ada juga brush yang terhubung dengan suara yang bereaksi pada setiap goresan brush. Berikut ini adalah contoh tampilan corel painter 2016 :



(sumber:<https://dailysocial.id/post/corel-painter-2016-hadirkan-fitur-melukis-dengan-suara>)

2. Adobe Photoshop

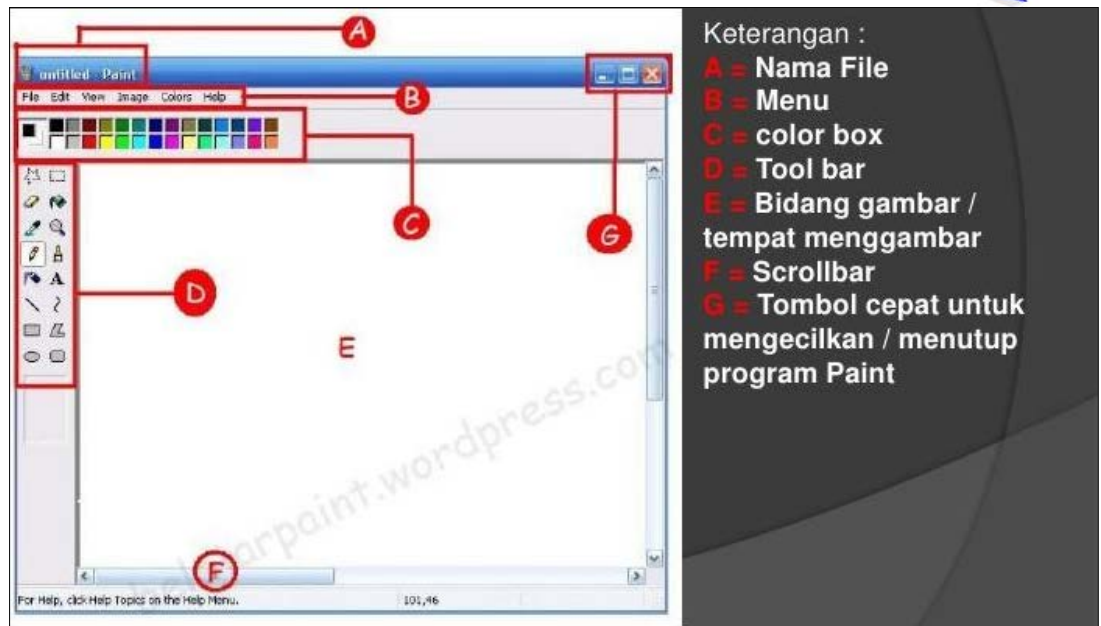
Adobe Photoshop merupakan tool perangkat lunak untuk mengolah citra digital. Adobe photoshop dikembangkan pada tahun 1987 dan memiliki lisensi Adobe System Incorporated pada tahun 1988. Standar industri aplikasi raster juga menggunakan aplikasi ini sebagai media pengeditan. Berikut ini adalah contoh tampilan halaman utama adobe photoshop :



(Sumber : <https://www.infopubliknews.com/2020/06/bagian-bagian-adobe-photoshop.html>)

3. Microsoft Paint

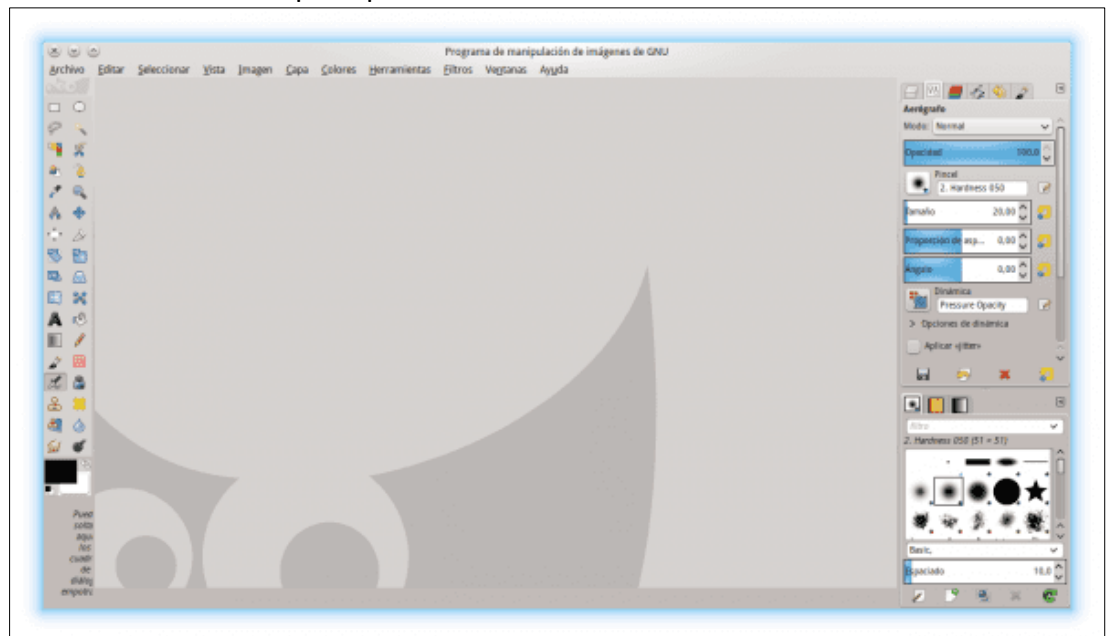
Microsoft paint merupakan perangkat lunak milik Microsoft yang menyediakan fasilitas pengolahan citra digital yang cukup lengkap seperti membuat garis lurus, membuat garis melengkung, mengedit teks, memberi warna, dll. Tampilan Microsoft Paint dapat dilihat sebagai berikut :



(Sumber : <https://pt.slideshare.net/Et3nK/mengenal-microsoft-paint>)

4. GIMP

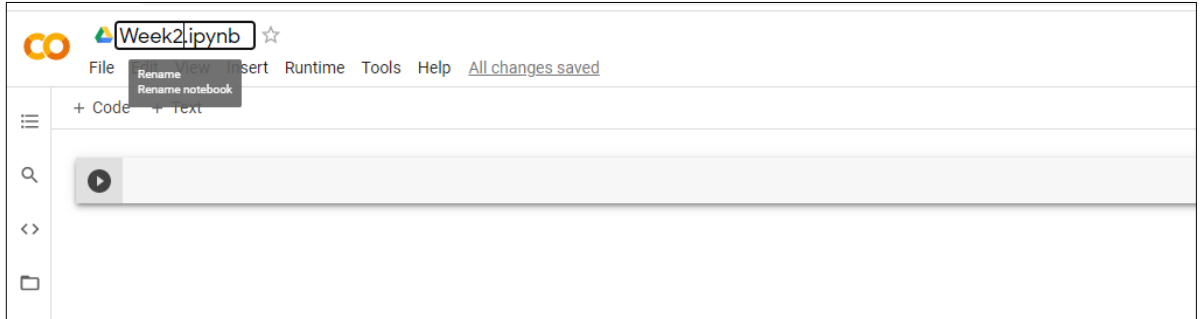
GNU Image Manipulation Program (GIMP) merupakan perangkat lunak pengolahan citra. GIMP berjalan pada desktop GNOME. GIMP memiliki kemiripan dengan fungsi-fungsi yang mirip dengan Photoshop. GIMP merupakan perangkat lunak gratis yang memiliki kemampuan seperti dekomposisi gambar, konversi gambar, membuat gambar berskala besar, dll. Berikut ini adalah tampilan pada GIMP.



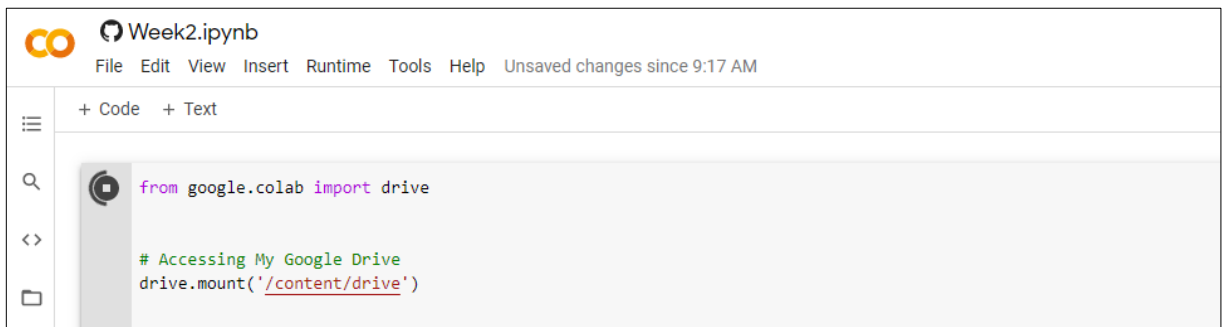
(Sumber : <https://blog.desdelinux.net/id/beri-gimp-tampilan-photoshop-cs6/>)

D. PRAKTIKUM

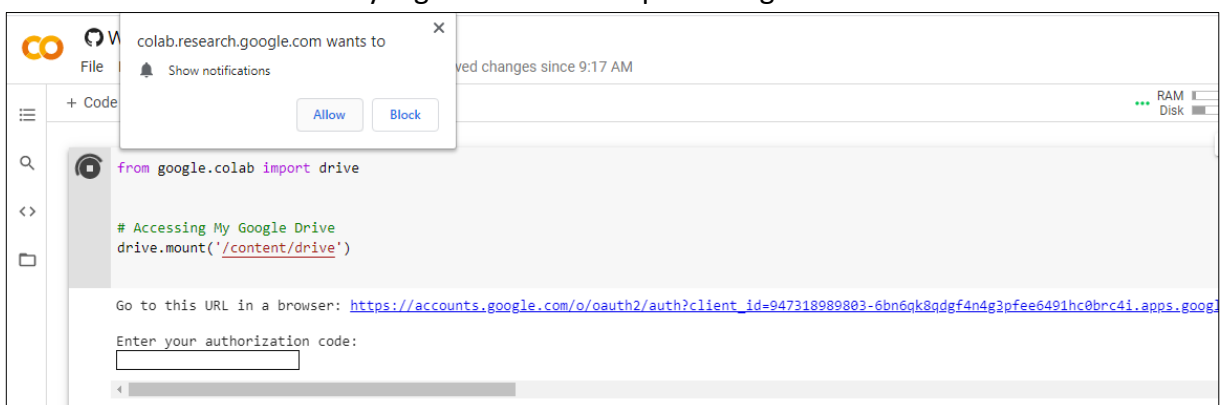
1. Buka <https://colab.research.google.com/>. Setelah dipastikan bahwa google Colab terhubung dengan Github Anda, lanjutkan dengan memilih repository yang telah digunakan pada praktikum minggu lalu, rename file menjadi “Week2.ipynb”.



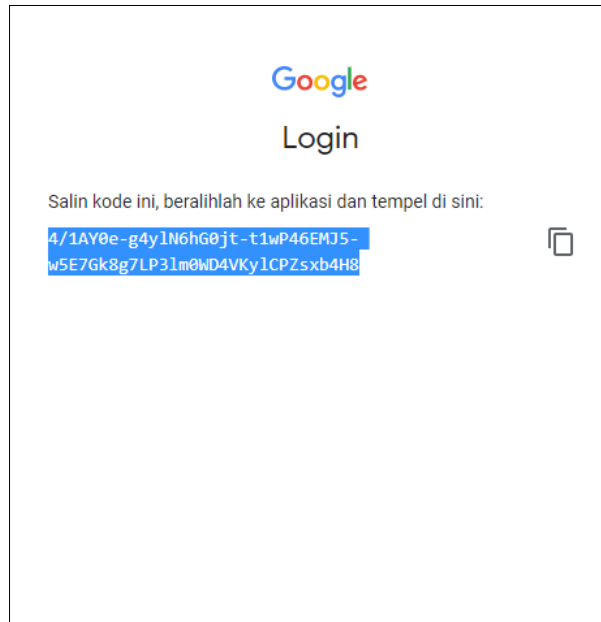
Kemudian import folder yang ada di Drive Anda dengan cara sebagai berikut.



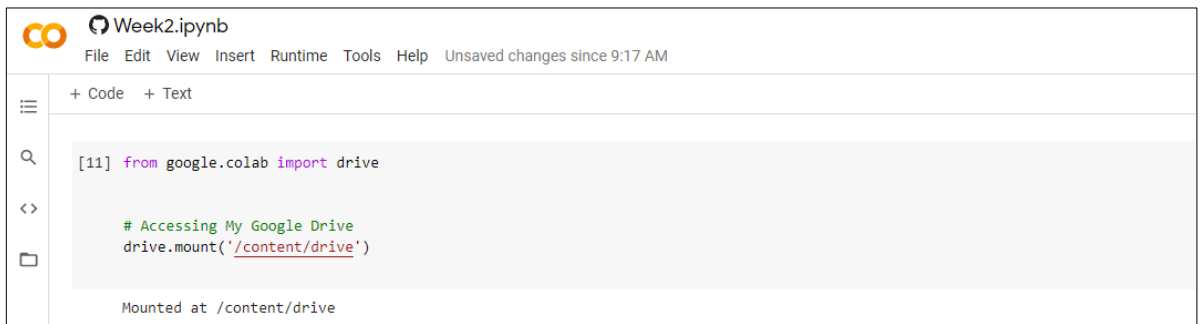
Proses sinkronisasi google colab untuk terhubung dengan gdrive membutuhkan sedikit perubahan setting notifikasi seperti gambar berikut. Kemudian akan muncul suatu URL yang akan mengarah ke new tab untuk login akun google. Pada tahap ini dibutuhkan kode otorisasi yang muncul setelah proses login dilakukan.



Setelah proses login dilakukan, maka salin kode otorisasi dan tempelkan pada field yang sudah disediakan pada google Colab.



Jika berhasil, maka Google Colab sudah dapat mengakses folder gdrive Anda dengan keterangan output program “Mounted at /content/drive”.



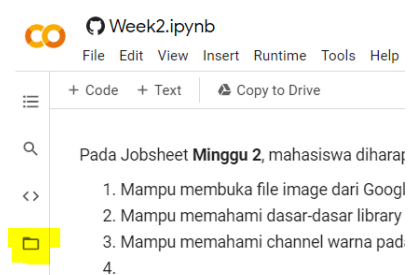
```

[11] from google.colab import drive

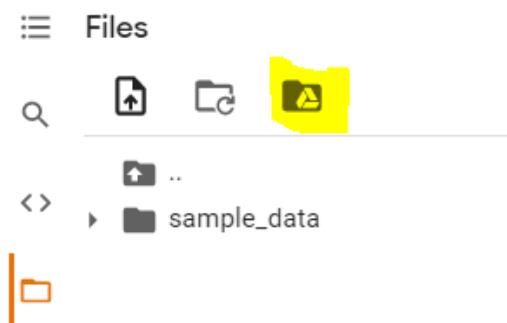
# Accessing My Google Drive
drive.mount('/content/drive')

Mounted at /content/drive
    
```

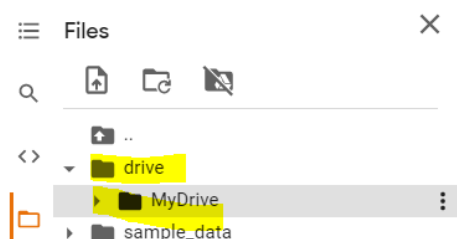
Cara yang kedua dengan mengklik tombol folder di sebelah kiri



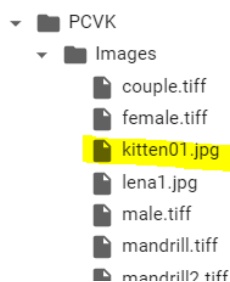
Pilih tombol gdrive -> Connect to Google Drive



Drive akan muncul dalam list folder



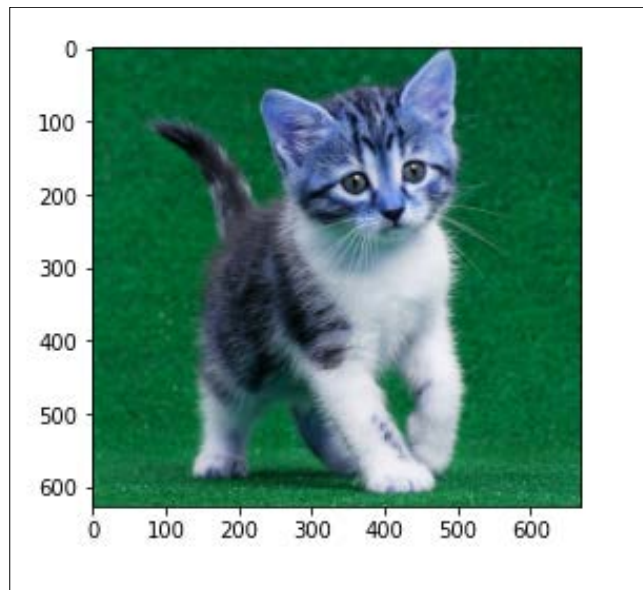
Pilih file dan klik icon titik 3 disebelah kanan file tersebut. klik copy path:



Kemudian kode program dapat dilanjutkan dengan membuka file Google Drive yang sudah ada. Contohnya pada kode program di bawah file image dengan nama kitten01.jpg akan dibuka untuk diproses lebih lanjut.

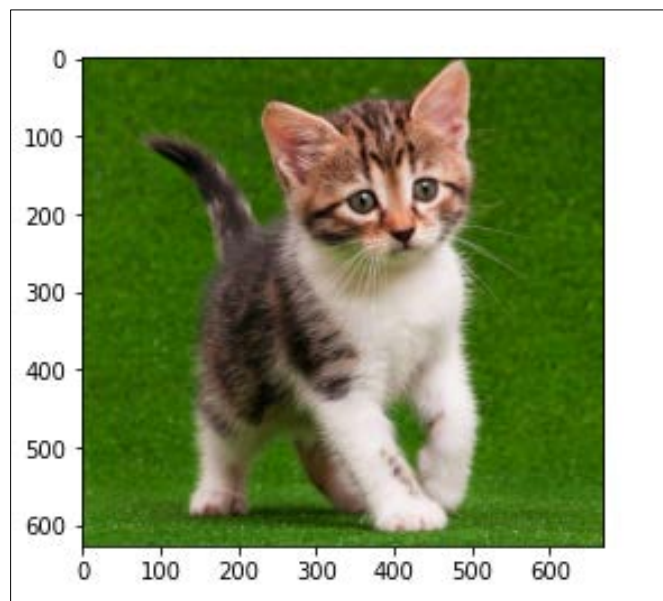
```
import cv2 as cv
from google.colab.patches import cv2_imshow
from skimage import io
import matplotlib.pyplot as plt
import numpy as np
img = cv.imread('/content/drive/MyDrive/PCVK_MM/Images/kitten01.jpg')
plt.imshow(img)
#perhatikan hasilnya adalah citra dgn channel warna BGR
```

Hasil nya berupa output gambar yang sudah di plot dengan matplotlib untuk mengetahui ukuran panjang dan lebar dari gambar tersebut.



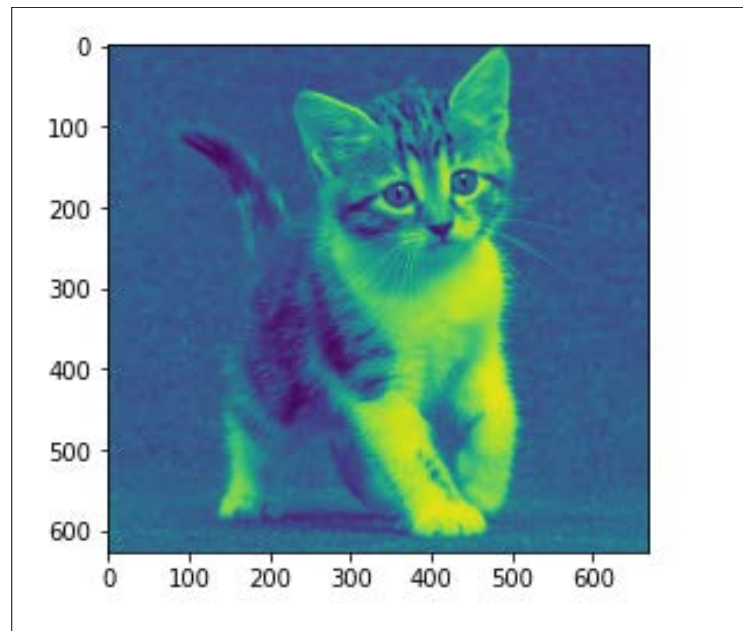
2. OpenCV membaca image dan menyimpan dalam channel warna BGR (Blue Green Red)
Tampilkan nilai pixel dan image

```
img2 = img  
img3 = cv.cvtColor(img, cv.COLOR_BGR2RGB)    #konversi channel BGR -> RGB  
plt.imshow(img3)
```



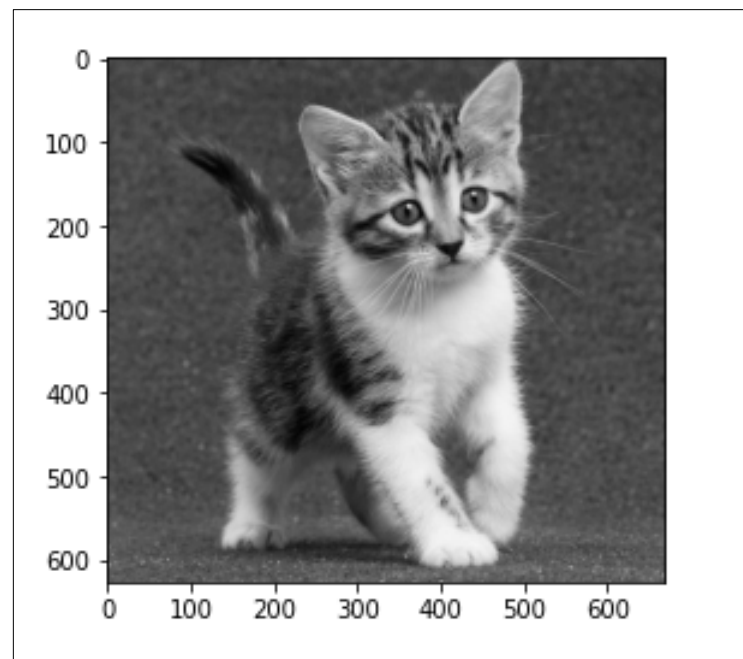
3. Menampilkan citra Grayscale, melakukan resizing, melakukan Flipping, Menyimpan citra hasil

```
img_gray = cv.imread('/content/drive/MyDrive/PCVK_MM/Images/kitten01.jpg', cv.IMREAD_GRAYSCALE)  
plt.imshow(img_gray)
```



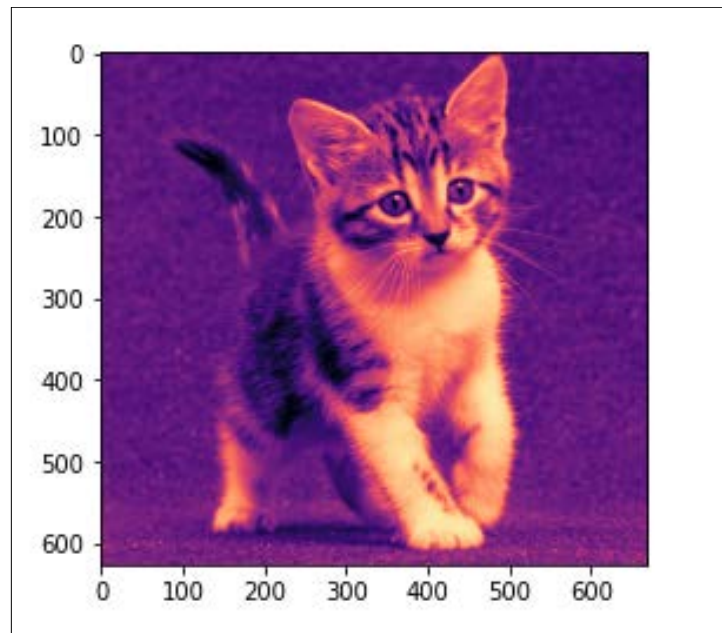
Dari citra grayscale, selanjutnya dipilih untuk ditampilkan colormap dengan warna 'gray'

```
plt.imshow(img_gray, cmap='gray')
```



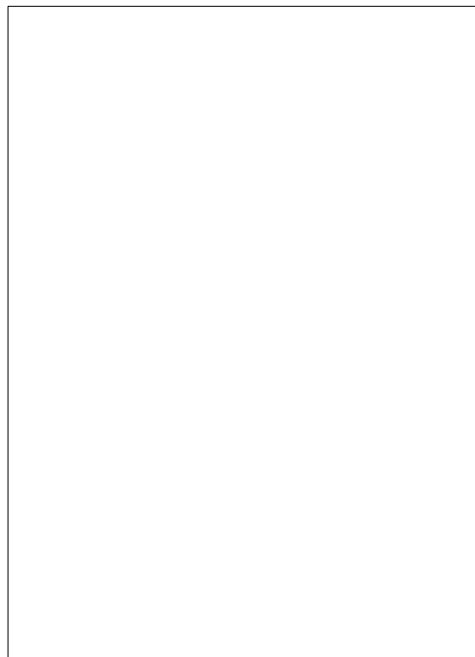
Dari citra grayscale, selanjutnya dipilih untuk ditampilkan colormap dengan warna 'magma'

```
plt.imshow(img_gray, cmap='magma')
```



Citra RGB di resize dalam ukuran panjang 512 dan lebar 1024

```
img4 = cv.resize(cv.cvtColor(img,cv.COLOR_BGR2RGB), (512,1024))  
plt.imshow(img4)
```



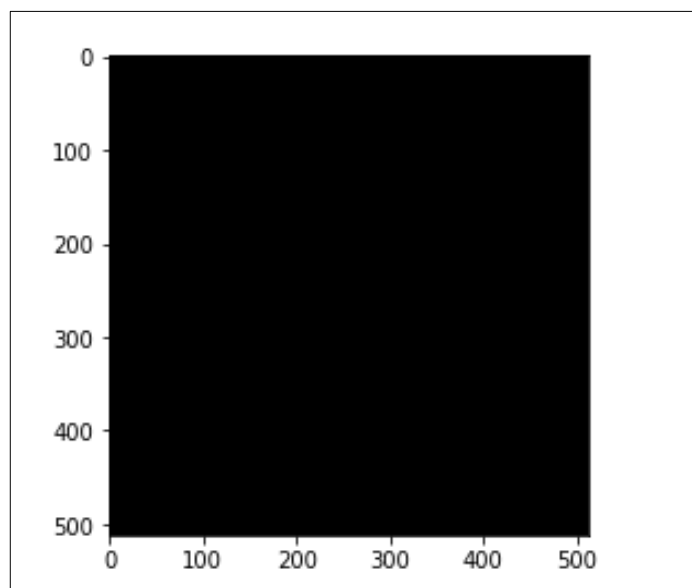
Citra RGB ditampilkan dalam ukuran yang lebih besar dengan posisi gambar terbalik

```
img5 = cv.flip(cv.cvtColor(img,cv.COLOR_BGR2RGB),0)  
  
#simpan File image  
#cv.imwrite('/content/drive/MyDrive/PCVK_MM/Images/mandrill2.tiff',img5)  
  
#tampilkan plot dengan ukuran canvas yg lebih besar  
fig = plt.figure(figsize=(10,10)) #ubah-ubah ukuran (10,10) sesuai kebutuhan  
ax = fig.add_subplot(111)  
ax.imshow(img5)
```



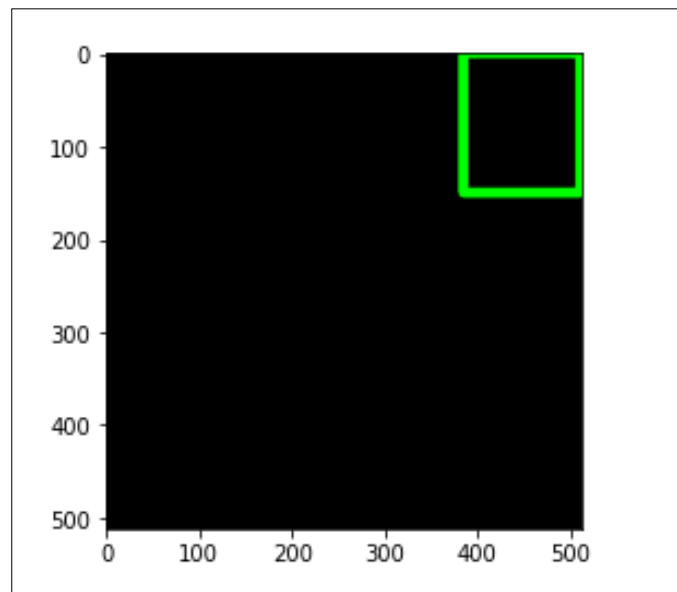
4. Membuat bentuk Geometri 2D dari OpenCV. Diawali dengan pembuatan black image dengan tipe data int16.

```
black_img = np.zeros(shape=(512,512,3),dtype=np.int16)  
plt.imshow(black_img)
```



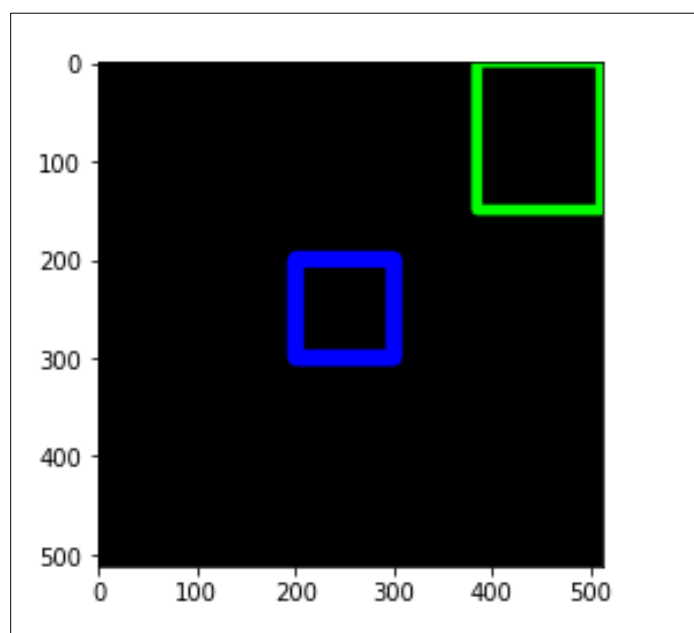
Kemudian menambahkan bentuk persegi panjang sesuai koordinat pt1 dan pt2

```
#perhatikan koordinat titik2 pt1 dan pt2
cv.rectangle(black_img,pt1=(384,0),pt2=(510,150),color=(0,255,0),thickness=10)
plt.imshow(black_img)
```



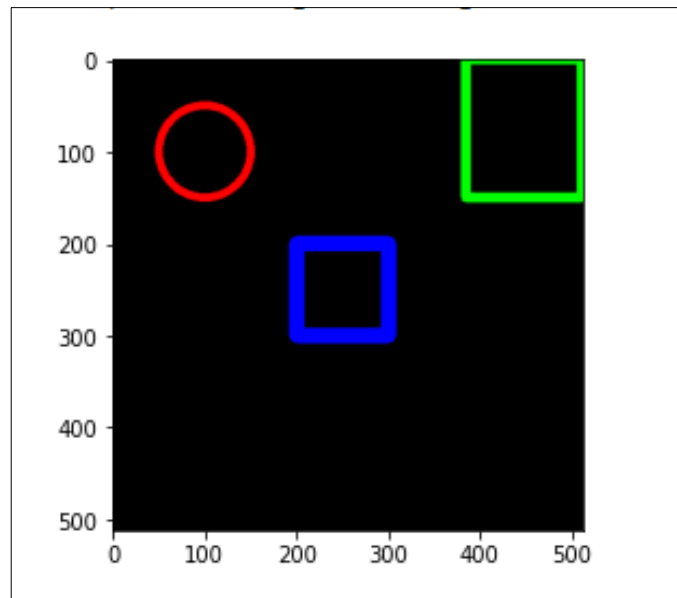
Selanjutnya ditambah menambahkan bentuk persegi sesuai koordinat pt1 dan pt2 yang tertulis pada kode program.

```
cv.rectangle(black_img,pt1=(200,200),pt2=(300,300),color=(0,0,255),thickness=15)
plt.imshow(black_img)
```



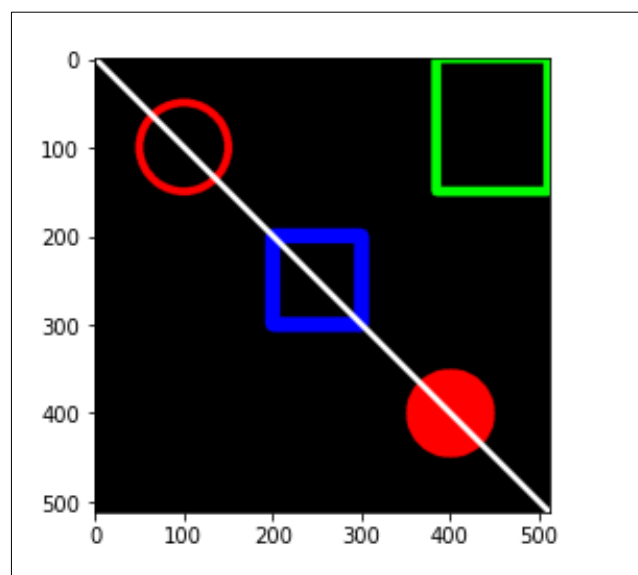
Tahap selanjutnya ditambah menambahkan bentuk lingkaran sesuai radius yang tertulis pada kode program.


```
cv.circle(black_img,center=(100,100),radius=50,color=(255,0,0),thickness=8)
plt.imshow(black_img)
```



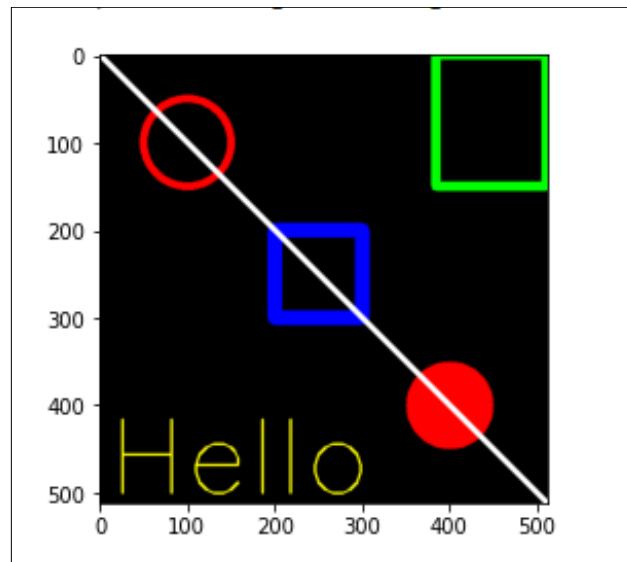
Kemudian dilakukan penambahan garis sesuai koordinat pt1 dan pt2 sebagai berikut.

```
cv.line(black_img,pt1=(0,0),pt2=(512,512),color=(255,255,255),thickness=5)
plt.imshow(black_img)
```



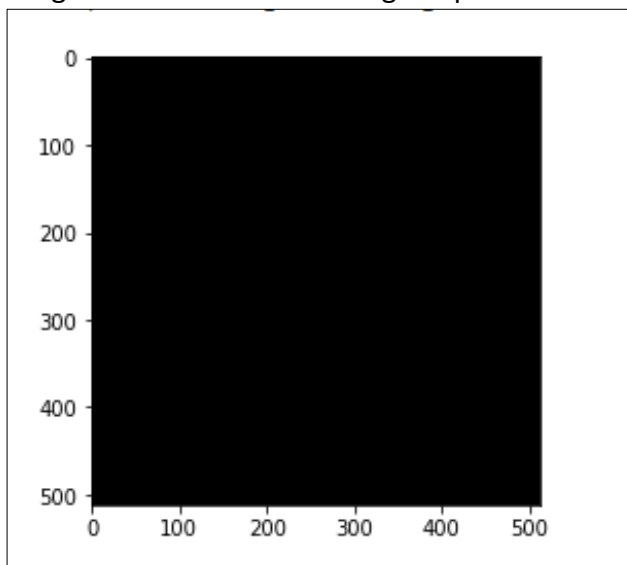
Penambahan text dengan font yang telah tertulis dengan ukuran yang sudah ditentukan.

```
font = cv.FONT_HERSHEY_SIMPLEX
cv.putText(black_img,text='Hello',org=(10,500),fontFace=font,fontScale=4,color=(255,255,0),thickness=2,lineType=cv.LINE_AA)
plt.imshow(black_img)
```



```
black_img2=np.zeros(shape=(512,512,3),dtype=np.int32)
plt.imshow(black_img2)
```

Pembuatan black image kembali dilakukan dengan tipe data int32



Berikut adalah kode program untuk inisialisasi NumPy array dengan tipe data int32

```
vertices = np.array([[100,300],[200,200],[400,300],[200,400]],dtype=np.int32)
vertices
```

```
array([[100, 300],
       [200, 200],
       [400, 300],
       [200, 400]], dtype=int32)
```

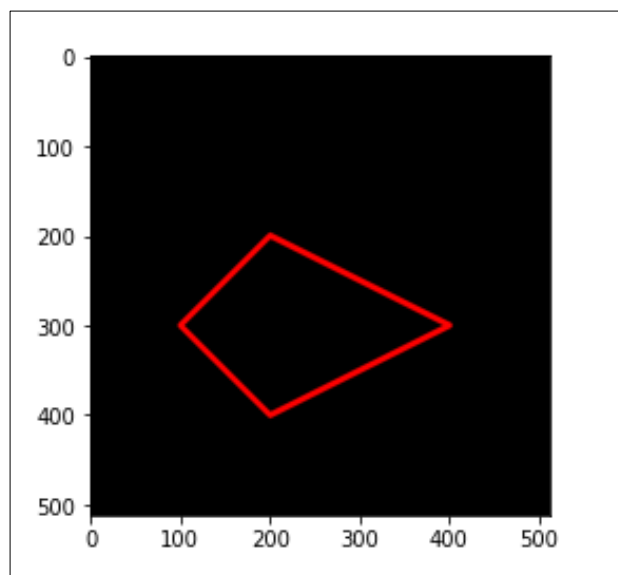
Array tersebut kemudian di reshape sebagai berikut

```
pts = vertices.reshape((-1,1,2)) # nilai 2 untuk menunjukkan bahwa tiap titik dibuat 3 channel yg mewakili R, G, dan B
pts
```

```
array([[100, 300]],
      [[200, 200]],
      [[400, 300]],
      [[200, 400]]], dtype=int32)
```

Penambahan polyline pada black image kedua yang telah dibuat.

```
cv.polylines(black_img2,[pts],isClosed=True,color=(255,0,0),thickness=5)
plt.imshow(black_img2)
```



- Setelah semua kode selesai simpan “Week2.ipynb” pada GitHub Anda dengan memilih File kemudian “Save a copy in GitHub”.

Pertanyaan

- Apakah perbedaan gambar yang ditampilkan tanpa dan dengan matplotlib?
- Apakah perbedaan dan pengaruhnya pembuatan black image antara tipe data int16 dan int32?
- Apakah kegunaan “google.colab.patches import cv2_imshow” pada potongan kode berikut

```
from google.colab.patches import cv2_imshow
from skimage import io
```

- Apakah kegunaan “skimage import io” pada potongan kode soal nomor 3

E. TUGAS

Berdasarkan praktikum bagian 1 dan 2 kerjakan beberapa tugas berikut :



1. Dengan menggunakan figsize, perhatikan apakah ukuran image pixelnya juga berubah?
2. Tampilkan image dalam channel Red-Green dan Green-Blue saja!
3. Tampilkan image baris ke 10-100, kolom 10-100!
4. Tampilkan image baris ke 5-30, semua kolom, channel Green saja!
5. Buat 5 kotak berbagai ukuran dan warna yang berbeda dalam satu image. disarankan menggunakan bilangan acak/random!
6. Tampilkan image dengan posisi terbalik!

Berdasarkan praktikum bagian 3 dan 4 kerjakan beberapa tugas berikut :

7. Buat rectangle dan circle pada bagian wajah dari image foto anda saat beraktifitas (bukan pasfoto).
8. Buat rectangle pada bagian sudut bawah kiri channel B pada color space RGB dari citra kitten/ lena/ mandrill/ male/ female/ couple/ sailboat/ peppers!
9. Lengkapi tulisan nama file pada file citra dari soal no.8. gunakan font, ukuran font, dan warna font yang sesuai keinginan anda.
10. Tunjukkan code program anda pada bapak/ibu dosen

--- SELAMAT BELAJAR ---