MODULE 1 – Using GitHub, Google Collaboratory, Reading and Viewing Image

A. PURPOSE

1. Students are able to create *Repository* on *Github*
2. Students are able to create *Python Notebook* on *Google Collaborator*
3. Students are able to save and access *Python Notebook* from *Google Collaborator*
4. Students are able to open Image *Files* from *Python Code*
5. Students are able to access pixels in the image

B. TOOLS AND MATERIALS

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

C. Theoretical Background

Digital Image (Raster Image) is a numerical representation of a two-dimensional image. The numeric value represented is generally an 8-bit binary value. This binary value is stored in image elements which are often referred to as pixels. A digital image contains pixels with a fixed number of rows and columns. Pixel is the smallest image element of a digital image. Pixels are stored in computer memory as a raster map, a two-dimensional array of integer type.

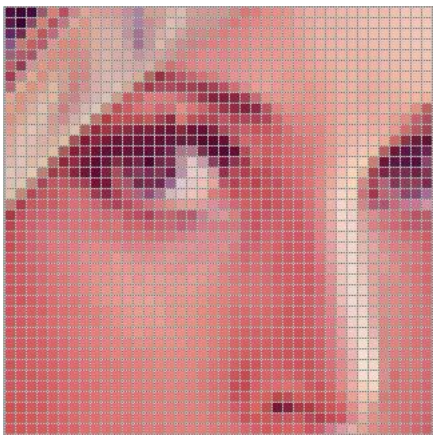


Figure 1. Images with pixel representation

Raster images are acquired using various input devices and techniques, such as digital cameras, scanners, radar, infrared cameras, and other. Digital Image Processing is a science that studies image transformation algorithms.

Digital images that are commonly used in image processing are binary, gray images, and RGB color images (Red, Green, and Blue).

Binary images are images that have only two colors, namely black and white. For example, if represented by an 8-bit binary value, black is worth 0000 0000, and white is 1111

1111. Usually displayed with normalized values 0 and 1, decimal 0 and 255, or hexadecimal 00x and FFx.

A gray image is an image that has 256 degrees of gray. Starting most negligible pixel value of black color, and the largest pixel value is white color. The following figure represents the normalization value of the gray image where 1.0 represents white, 0.0 represents black, and values between 0.0 – 1.0 represent the degree of a gray color.

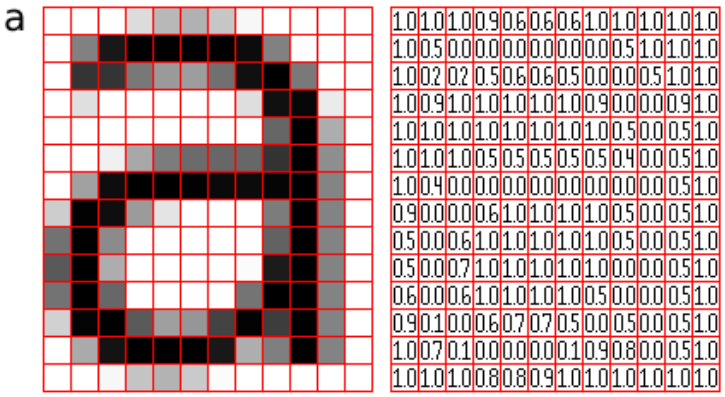


Figure 2. Grayscale image representation

RGB color image is an image that has three levels/color channels represented by a 3-dimensional image resolution. In digital images, the first level is used to store the color R (Red / red), the second level is used to store the color G (Green/green), and the third level is used to store the color B (Blue / blue). In output devices such as LCD monitors, RGB is arranged in an array and is very small in size. The value representation is stored in the binary value. Black color is represented by R = 0000 0000, G = 0000 0000, and B = 0000 0000. Red color is represented by R = 1111 1111, G = 0000 0000, and B = 0000 0000. Because it has 3 levels, the file size is RGB. Compared to the grayscale image is 3:1 with the same number of pixels (image resolution). Colors other than red, green, and blue are mixtures of these three colors. The following figure shows; the yellow color is a mixture of red and green. Yellow is represented by the value R = 1111 1111, G = 1111 1111, and B = 0000 0000.

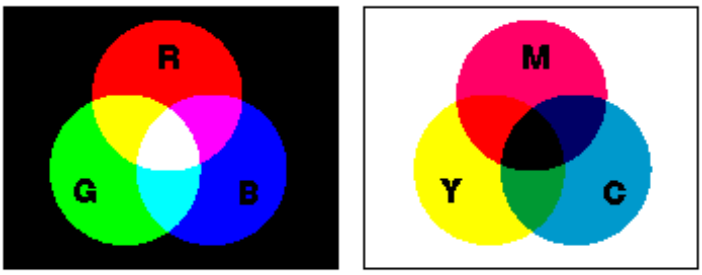


Figure 3. RGB and CMY color representation

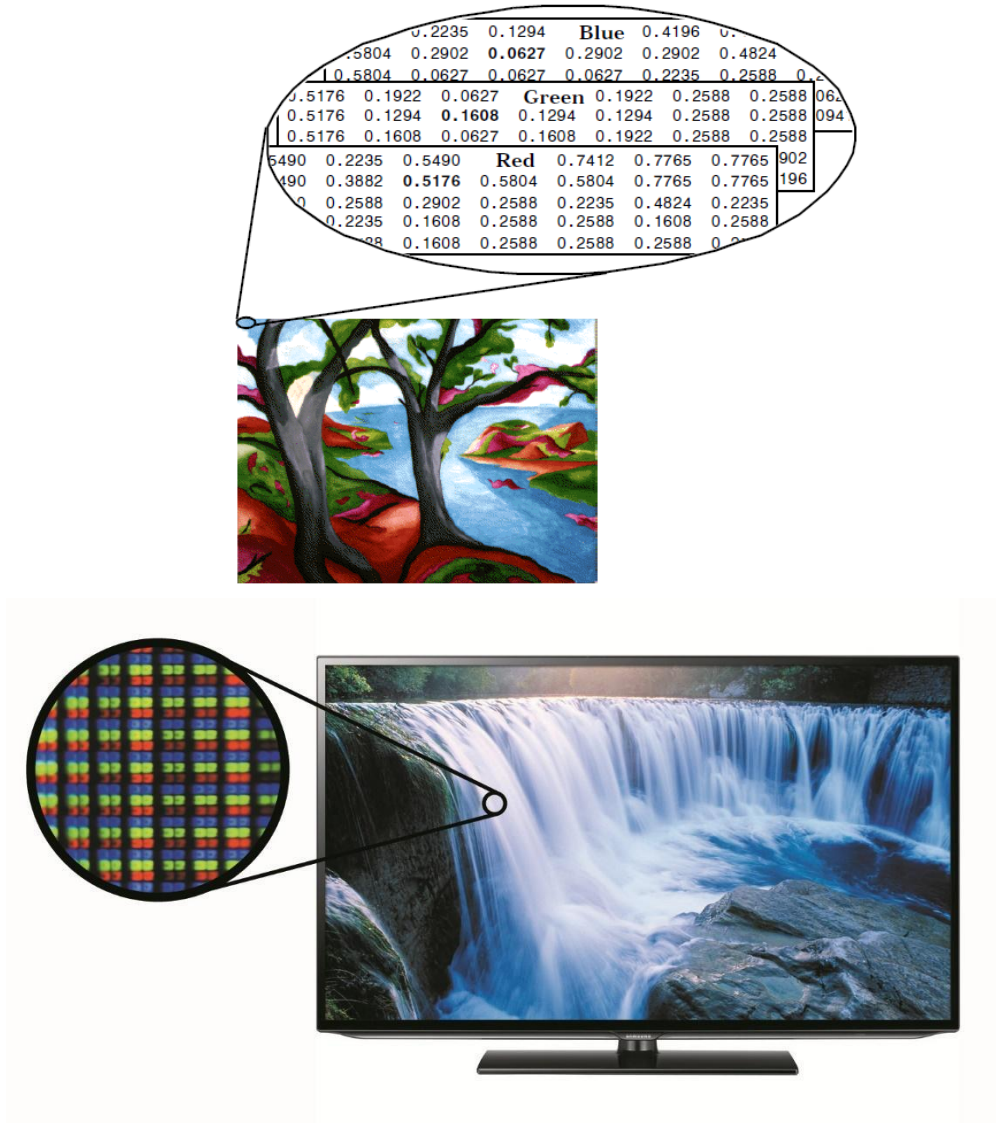


Figure 4. Digital and analog RGB representation

Digital images can be displayed on a variety of image viewer software (Image Viewer). Web Browsers can display standard Internet format images directly, such as GIF, JPEG, and PNG. In addition, some browsers can show SVG, which is the W3C standard format. Some current science images can be huge (for example, the 46 gigapixel image size of the Milky Way galaxy, measuring 194 GB).

Python is one of the trending programming languages in recent years. In the beginning, Python is a continuation of the ABC programming language developed by Guido van Rossum in 1990 at CWI, Amsterdam. Python has unique characteristics compared to other languages in writing program code that has rules regarding indentation, data types, tuples, and dictionaries. The advantages of Python compared to other programming languages are mainly seen in handling modules and the benefits of Python, an open-source, accessible, and multiplatform product. Python has several advantages: modules that Python has provided; grammar is easier to understand; owned layouts are easier to review and develop; object-oriented; memory processing is done in a modular manner; can be built in other languages Python itself or C/C++. The modules can be built in Python or C/C++, as well as other advantages.

Google Collaboratory or Google Colab is a cloud-based and free tool. Google Colab, made with the Jupyter environment and supports multiple libraries (libraries) are needed in the development environment of Artificial Intelligence (AI). Google Colab allows users to write and

execute Python in browsers (Chrome, Firefox, and Safari) without requiring configuration, access the GPU for free, and share program code (team collaboration) quickly. Google Colab can execute, write, store and even share program code that has been created via Google Drive. Writing Python program code and running the program code does not require installing the installer on the computer. Installation does not need to be done because all processes are done in the cloud. In addition, additional Python functionality can also take advantage of the built-in libraries that exist in Google Colab.

D. PRACTICUM

1. For those who don't have a Github account, you can create a new account on Github (https://github.com/join?ref_cta=Sign+up&ref_loc=header+logged+out&ref_page=%2F&source=header-home)

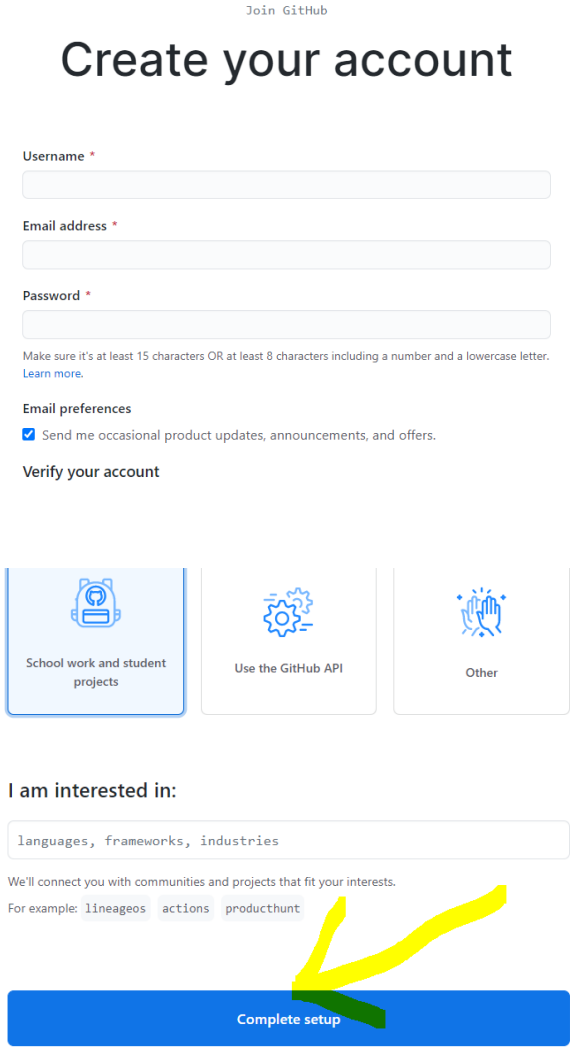


Figure 5. GitHub Registration Form

After filling in all the *mandatory fields* , click " *complete setup* ". The next step is to verify the email address used to create the new account.

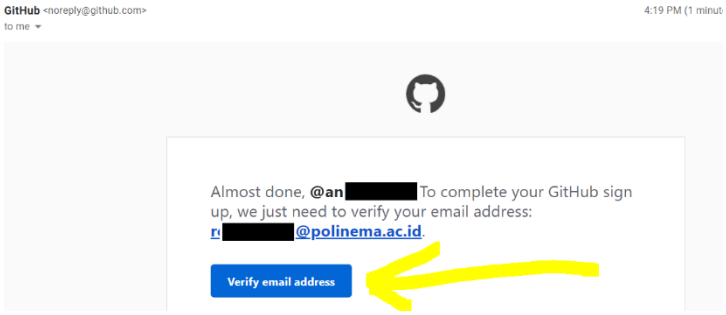


Figure 6. GitHub Account Email Registration Verification

2. Log into the created github account, and create a new repository, selecting " *Create a Repository* ".

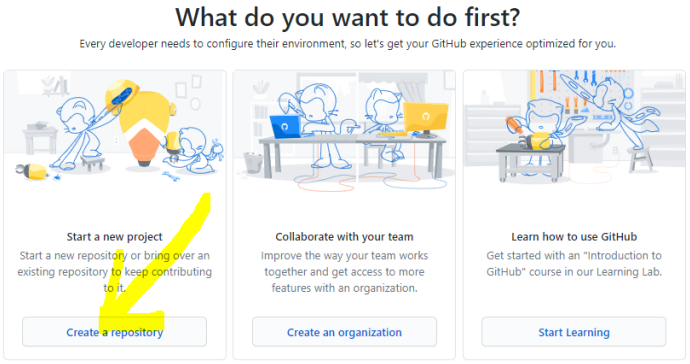


Figure 7. Create Repository GitHub

Fill in the repository name, description (optional), and choose whether the repository to be created is public or private . Select Private if the project does not want to be accessed by the public. After that click " Create repository " to create a new repository.

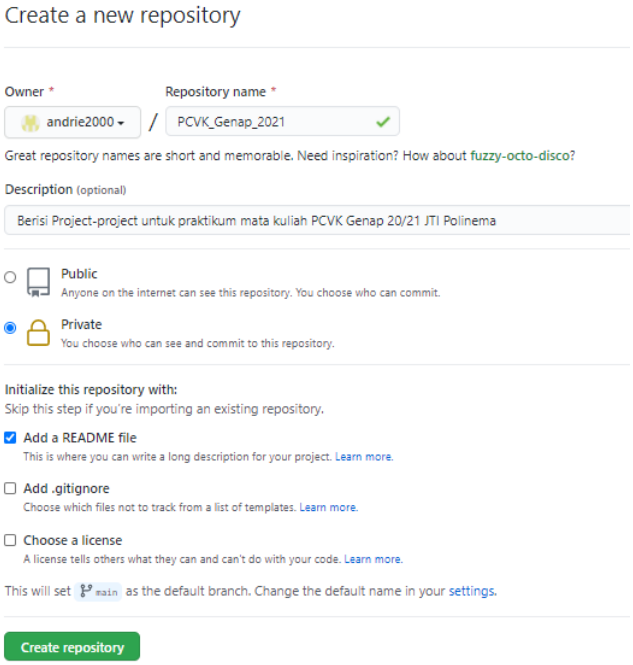


Figure 8. Creating Repository Window

3. The repository is ready to be filled with projects.

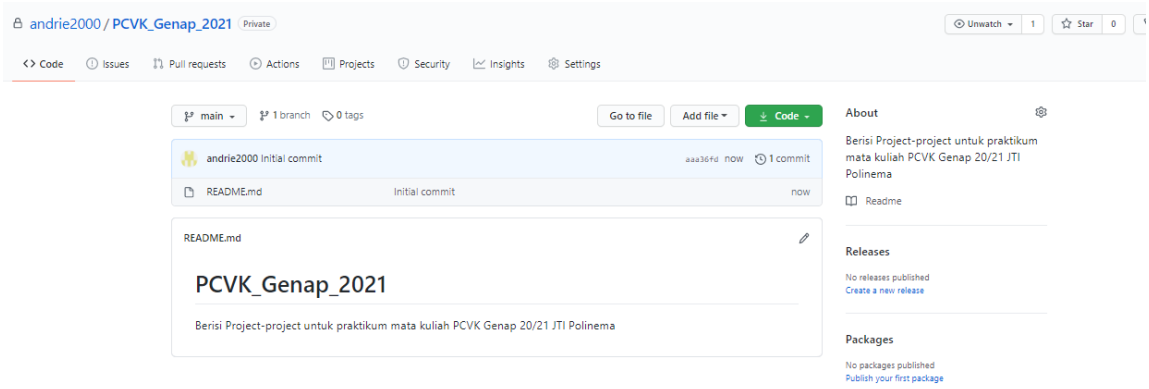


Figure 9. GitHub New Repository

In the image processing and computer vision practicum, we will do editing and build code for image processing using Google Collaboratory . Google Collaborator y can be searched using a search engine or can be directly opened at the following link: https://colab.research.google.com/

The display of the main window is as follows:

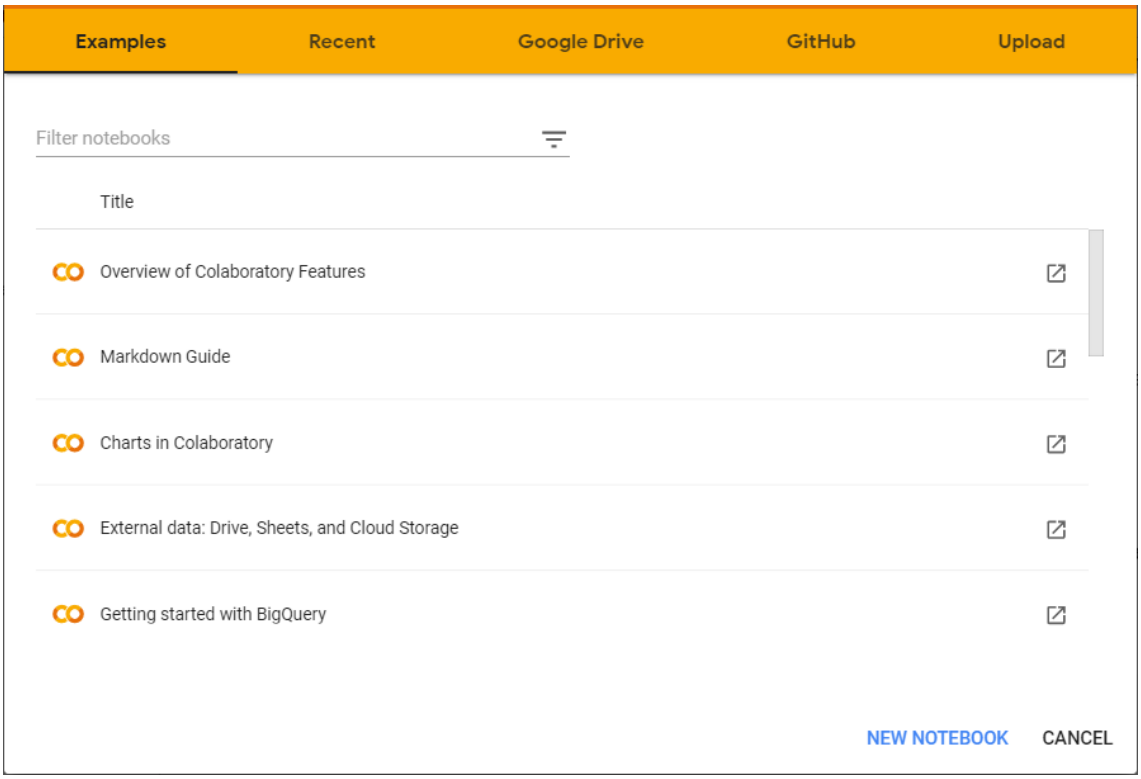


Figure 10. Google Collaborator Main Window

4. The new application will be created in python notebook format (file type is **.ipynb*), where the file can be stored on Google Colab cloud drives, Google Drive, or can be connected to Github directly. In this Lecture we will use G ithub. Select the Github menu in the main window, then proceed to connect with your personal Github.

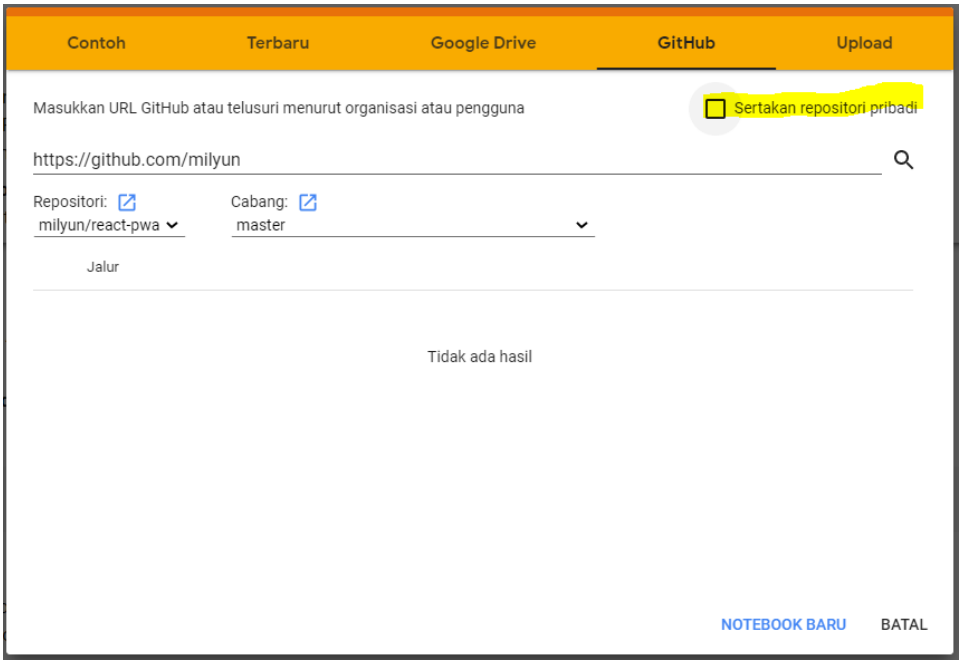


Figure 11. Google Colab connected wit GitHub Repository

Enter your GitHub account URL (https://github.com/account_name), then press enters or press the search icon, then a list of repositories and branches will appear in your account. If the storage created in step 2 is private, select the "Include private repository" checkbox, a window authorizing Google Colab will show, and then press the " Authorize googlecolab " button.

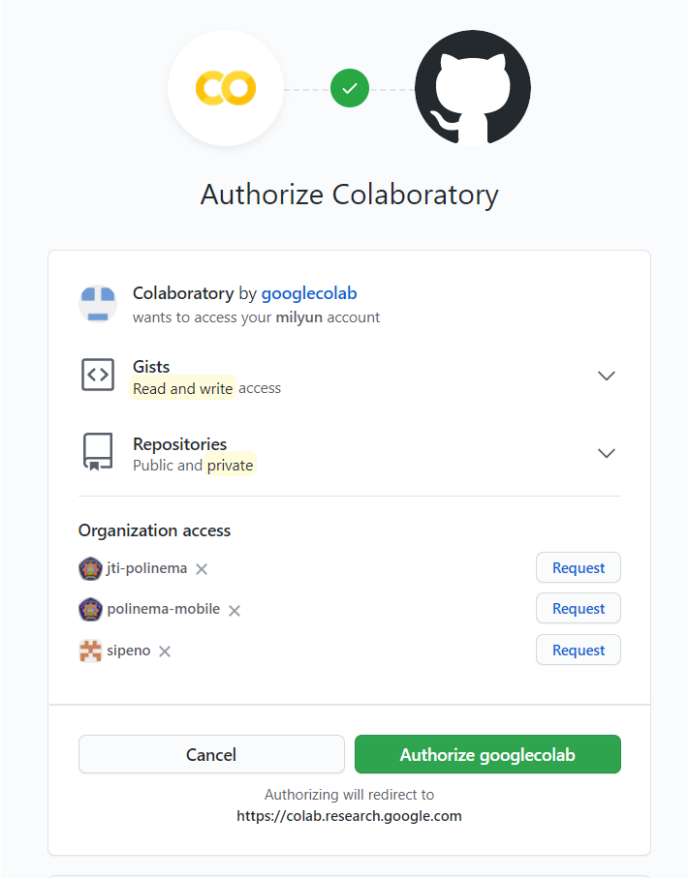


Figure 12. Authorizing Google Colab with GitHub

5. Once google colab is connected to Github, you can select the repository and create a new notebook.

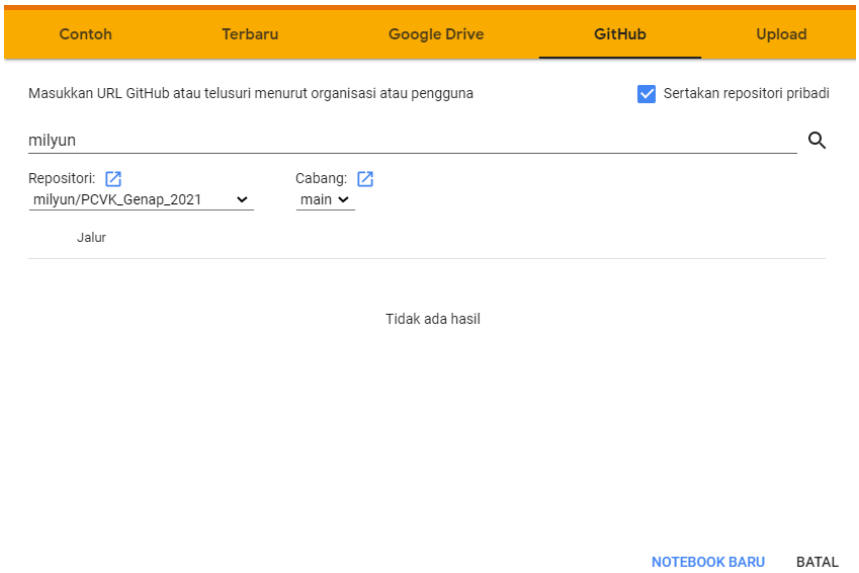


Figure 13. Google Colab already connected with GitHub

Rename the file (*.ipynb), then select File → Save Copies to Github

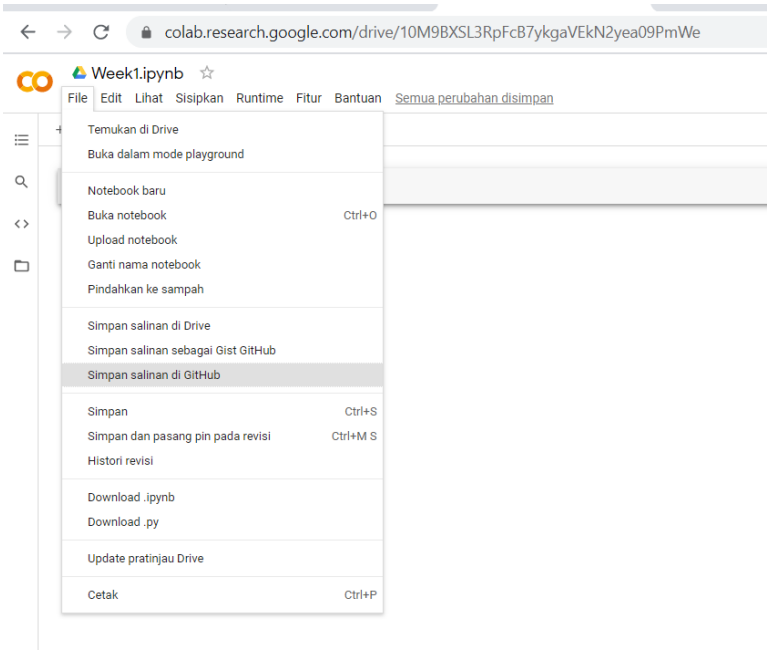


Figure 14. ipynb file saving to GitHub

Window of the following figure will show:

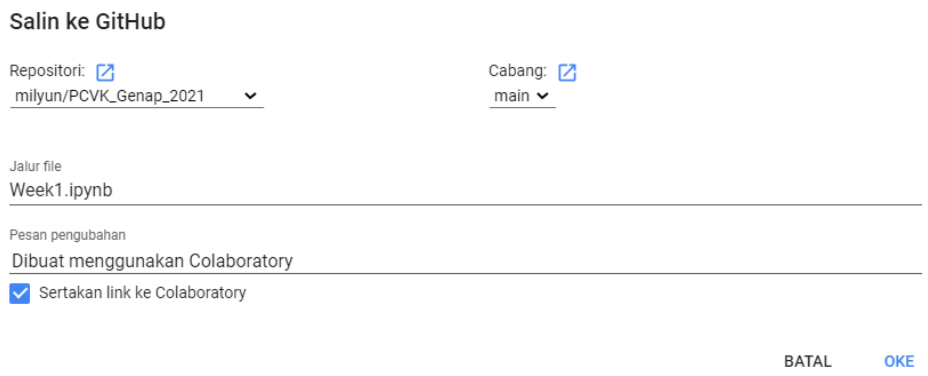


Figure 15. Saving File with filename filling in Google Collaborator

6. After pressing the "OK" button it will immediately open the notebook file page that was just created on the Github repository.

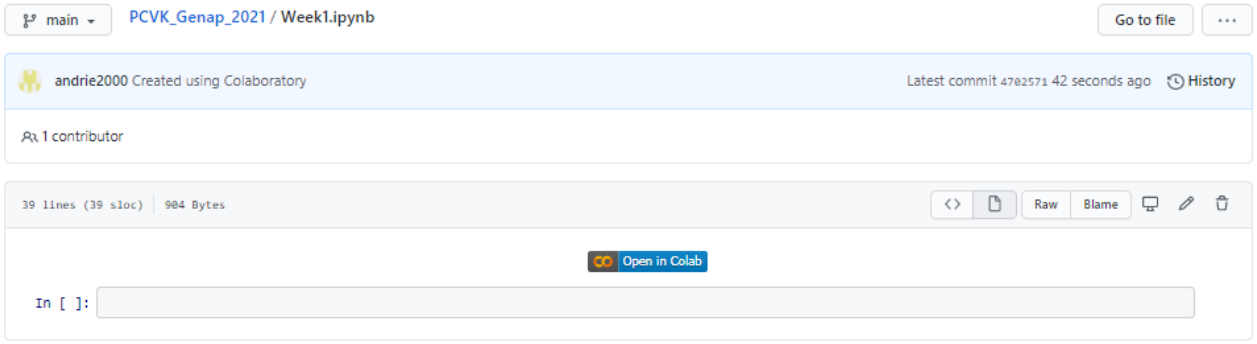


Figure 16. GitHub file preview from Google Collaborator

In the upper left corner of Google Colab, note that if the Githu b icon has appeared, it means that the file has been successfully accessed by Google Colab.

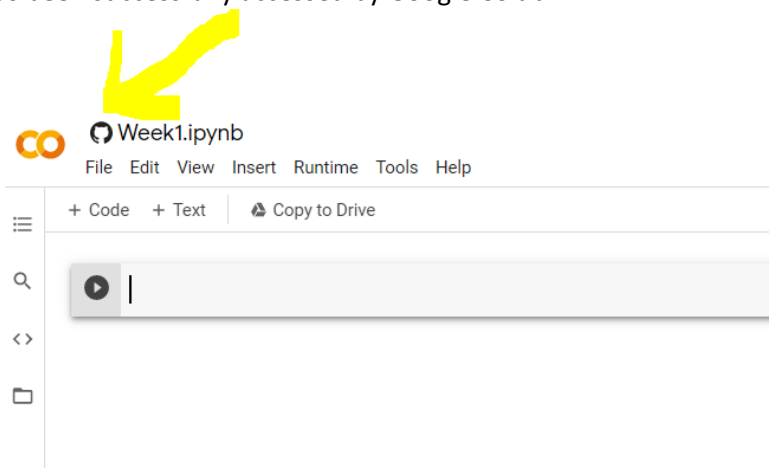


Figure 17. Google Collab File access from GitHub repository

Notebook files can contain formatted notes and contain python code that can be run directly. The " *+ Code* " button is used to add program code, and " *+ Text* " is used to add formatted notes.

7. Use the following libraries as a first step:

▾ Langkah (Step) 1:

Pada code awal, beberapa library yang akan digunakan adalah: **numpy, pandas, cv2, skimage, PIL, matplotlib**

--Bahasa--

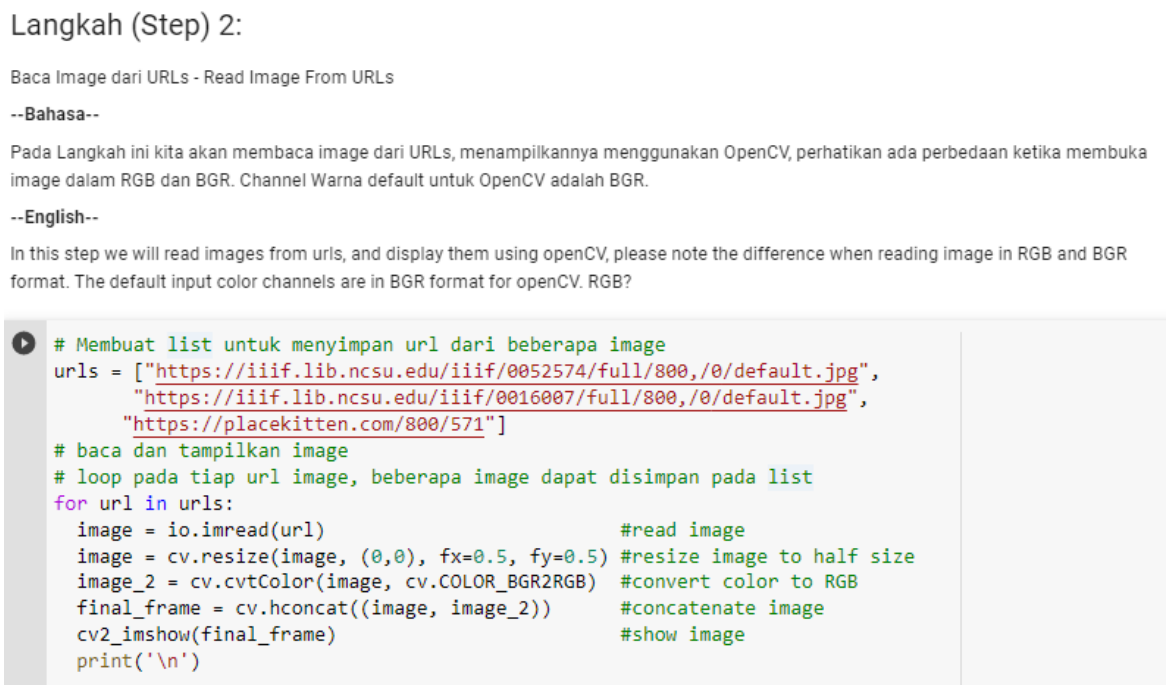
--English--

- Numpy is an array manipulation library, used for linear algebra, Fourier transform, and random number capabilities.
- Pandas is a library for data manipulation and data analysis.
- CV2 is a library for computer vision tasks.
- Skimage is a library which supports image processing applications on python.
- Matplotlib is a library which generates figures and provides graphical user interface toolkit.

```
[9] import numpy as np
    import pandas as pd
    import cv2 as cv
    from google.colab.patches import cv2_imshow # for image display
    from skimage import io
    from skimage import transform
    from PIL import Image
    import matplotlib.pylab as plt
```

Figure 18. Importing Library in Python

8. Step 2 is used to read and display the image



Langkah (Step) 2:

Baca Image dari URLs - Read Image From URLs

--Bahasa--

Pada Langkah ini kita akan membaca image dari URLs, menampilkannya menggunakan OpenCV, perhatikan ada perbedaan ketika membuka image dalam RGB dan BGR. Channel Warna default untuk OpenCV adalah BGR.

--English--

In this step we will read images from urls, and display them using openCV, please note the difference when reading image in RGB and BGR format. The default input color channels are in BGR format for openCV. RGB?

```
# Membuat list untuk menyimpan url dari beberapa image
urls = ["https://iiif.lib.ncsu.edu/iiif/0052574/full/800,/0/default.jpg",
        "https://iiif.lib.ncsu.edu/iiif/0016007/full/800,/0/default.jpg",
       "https://placekitten.com/800/571"]
# baca dan tampilkan image
# loop pada tiap url image, beberapa image dapat disimpan pada list
for url in urls:
  image = io.imread(url)                          #read image
  image = cv.resize(image, (0,0), fx=0.5, fy=0.5) #resize image to half size
  image_2 = cv.cvtColor(image, cv.COLOR_BGR2RGB)  #convert color to RGB
  final_frame = cv.hconcat((image, image_2))      #concatenate image
  cv2_imshow(final_frame)                         #show image
  print('\n')
```

Figure 19. Reading dan showing image in Python

In stage 2 we will create a list to store the URLs of several images, where for each image we will do: image reading, resizing the image size in half, converting color images into RGB format, combining the original image and the converted image, and the last is display the image. For the image url used, please copy and paste the following URL, or you can also use another image URL:

```
urls = ["https://iiif.lib.ncsu.edu/iiif/0052574/full/800,/0/default.jpg","https://iiif.lib.ncsu.edu/iiif/0016007/full/800,/0/default.jpg","https://placekitten.com/800/571"]
```

The result of the code snippet above is as follows:

Figure 20. Image showing result

9. Step 3 in this practicum is to see the image file size, in the following code:

Langkah (Step) 3

melihat ukuran File image

```
[51] tinggi = image_2.shape[0]
     lebar = image_2.shape[1]
     print("resolusi image: tinggi x lebar = ",tinggi," x ",lebar)
     cv2_imshow(image_2)
```

```
resolusi image: tinggi x lebar =  286  x  400
```

Figure 21. Showing Image Resolution

10. Step 3 in this practicum is to see the image file size, in the following code:

Langkah (Step) 4

mengakses pixel dengan memberi garis diagonal menyilang

```
image_2 = cv.cvtColor(image, cv.COLOR_BGR2RGB)
image_3 = cv.cvtColor(image, cv.COLOR_BGR2RGB)

#membuat garis horizontal ditengah image
for y in range (lebar):
  image_3[int((tinggi)/2),y] = [255,255,255]

final_frame = cv.hconcat((image_2, image_3))
cv2_imshow(final_frame)
```

Figure 22. Image with white diagonal line

Question

1. Explain why in this practicum module the execution of Python code is done using Google Colab?
2. Explain the use of each library in the eighth step practicum? Do all these libraries have to be used in this practical session?
3. In the 8th step of the experiment there is a snippet of program code as follows:
   ```
   image = cv.resize(image, (0,0), fx=0.5, fy=0.5)
   ```
   What is the use of the program code ? and what is the effect if it is not done?
4. Take a look at the following program code snippet:
   ```
   #membuat garis horizontal ditengah image
   for y in range (lebar):
     image_3[int((tinggi)/2),y] = [255,255,255]
   ```
   What is the use of the code [255,255,255] ? Explain!
5. Explain the relationship between pixels and high or low image resolution!

E. TASK

1. Do the practical steps as above
2. Create vertical lines and diagonal lines in the output image

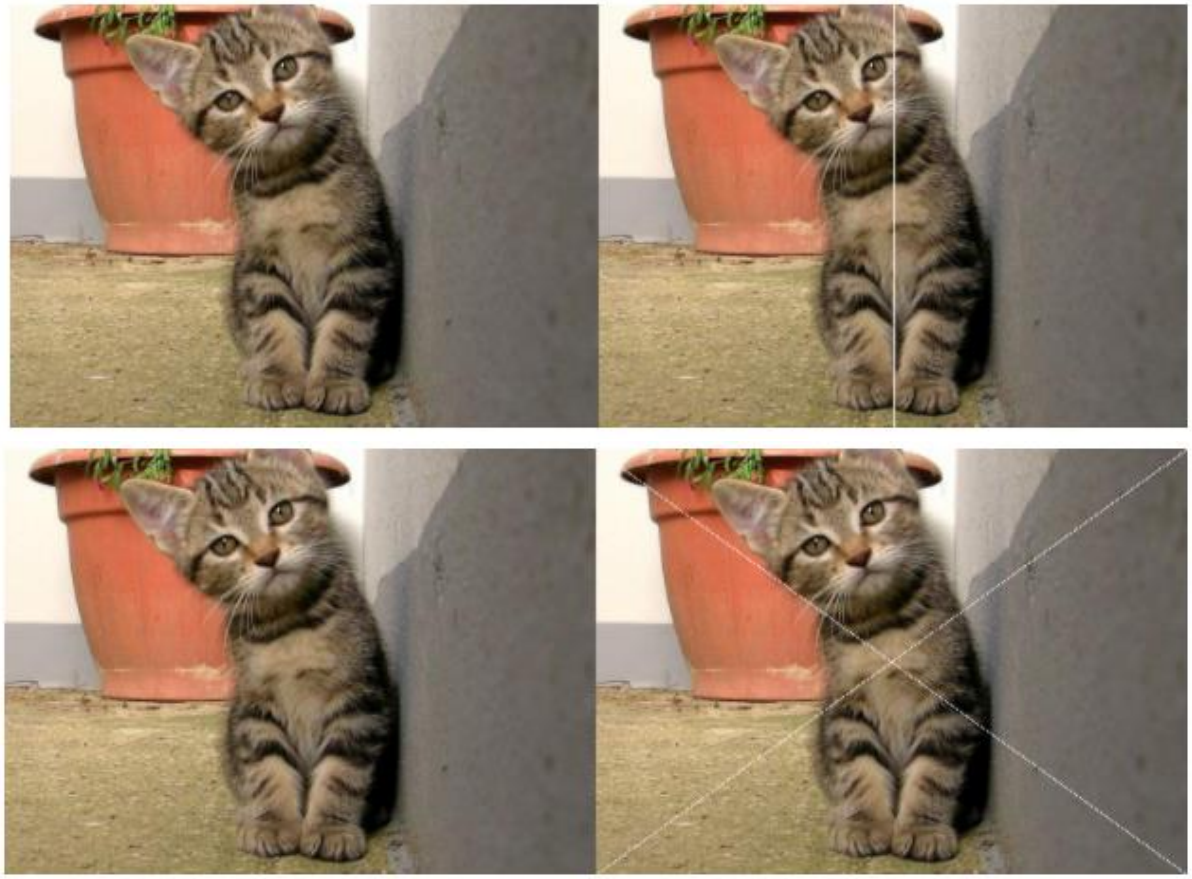


Figure 23. Image with Vertical Line and Diagonal Line

3. Draw the horizontal white lines in the middle of image with certain length

4. Draw a box from a collection of white pixel in certain area in the image

Form groups of 2-3 people each. For each group, do the following steps:

1. Access the local image file (KTP/SIM) and display it using OpenCV and Matplotlib
2. Identify why it uses BGR2RGB color conversion on matplotlib and no conversion on opencv
3. Cover each specific part of the identity card. Group 1 closes the NIK number and name section. Group 2 closes the TTL and Gender sections. Group 3 closes the Goals section. Blood and TTD. Group 4 closes the Address and District section. Group 5 closes the section on Religion and Marital Status. Group 6 close Occupation and nationality. Group 7 cover section applies to and photos. Group 8 cover the district section and the date the KTP was issued which is at the bottom of the photo. Group 9 closes the name of the province and district at the very top of the KTP. Group 10 closed the RT/RW and Kel/village sections. Cover using 2 boxes of different colors. Do not use the rectangle openCV function, this function will be studied in Week 2.

--- GOOD LUCK ---