Week 01

"Using GitHub, Google Collaboratory, Reading and Viewing Image"

Course Pengolahan Citra dan Visi Komputer

From :

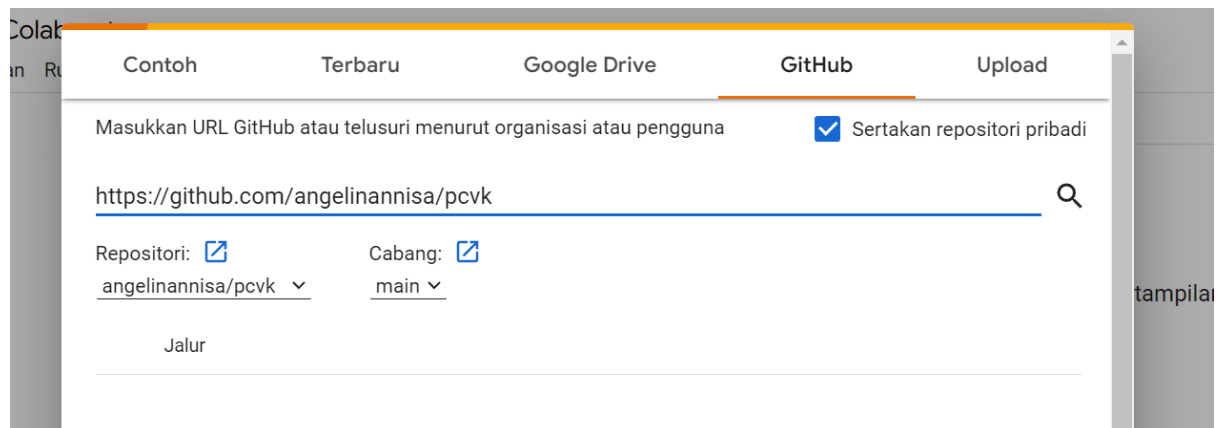Angelin Fitri Annisha (2341729007)

Mahasiswa PMM
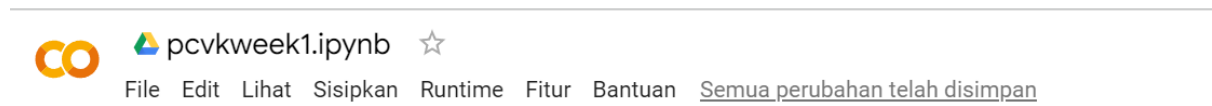
D-IV Informatics Engineering

Malang State Polytechnic
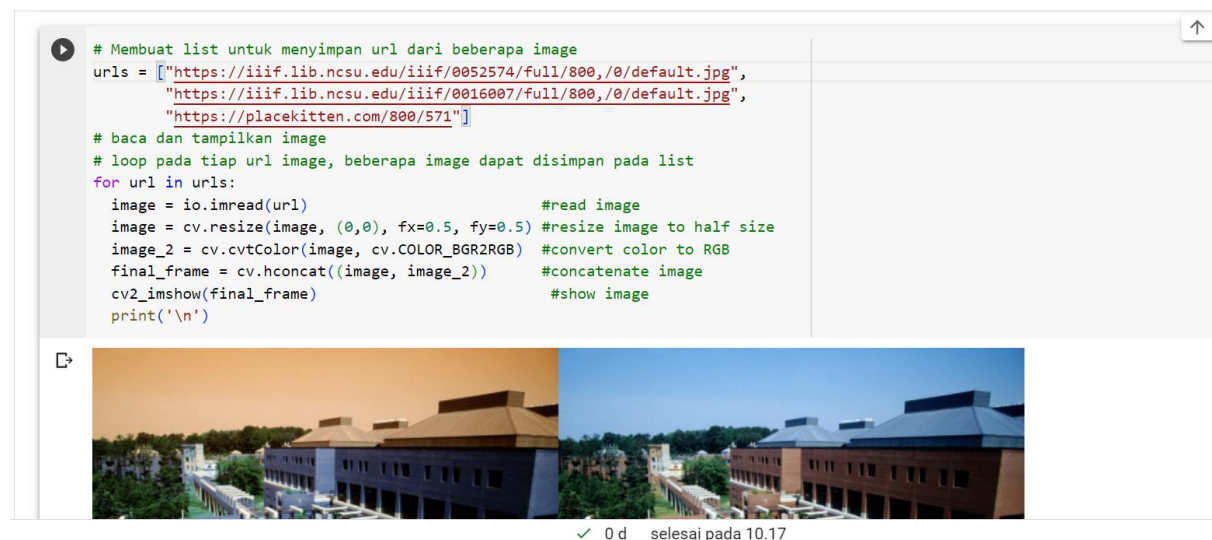
2023

1. PRACTICUM
- Connect the Google Colab to GITHUB

- Google Collab File access from GitHub repository

- Step 1

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

- Step 2

```
# 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')
```


✓ 0 d   selesai pada 10.17

- **Step 3**


```
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

+ Kode   + Teks

- **Horizontal lines**

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

for y in range (lebar):
  image_3[int((tinggi)/2),y] = [255, 255, 255]

  image_3[int((tinggi)/2),:] = [255,255,255]

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


✓ 0 d   selesai pada 10.17

2. QUESTIONS

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!

Answer:

1. Because Google Colab is a cloud-based platform that allows users to write and execute Python code in a web browser, without the need for any local installations. This makes it accessible to a wide range of users, including those who might not have powerful hardware or admin privileges to install software on their local machines.

2. I think it should be, if you don't use the library there will be a code program error.
- Numpy is an array manipulation library
- Matplotlib is a library for generating figures and providing a graphical user interface toolkit
- Pandas is a library for data manipulation and analysis

3. Make the image you want to display look neater with the programmed size then the image size will be messed up.

4. The code [255,255,255] is a notation commonly used in RGB (Red-Green-Blue) based images.In the RGB color model, colors are represented by three values the amount of red, green, and blue light that should be mixed together to create the desired color. Each component can take on values from 0 to 255, where 0 represents no intensity of that color, and 255 represents the maximum intensity.

5. the number of pixels in an image directly affects its resolution. Higher pixel counts lead to higher resolutions, which generally result in greater detail and image clarity. The choice of image resolution depends on the specific use case, as high-resolution images are essential for tasks where detail is critical, while low-resolution images are suitable when smaller file sizes or faster loading times are more important than detail.

3. TASK
Create vertical lines and diagonal lines in the output image

- Vertical lines

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

     for y in range (tinggi):
       image_3[y,int((lebar)/2)] = [255, 255, 255]

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

- Diagonal lines

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

for y in range (tinggi):
  image_3[y,int((lebar)/4)]

  cv.line(image_3, (400,0), (0,286), [255,255,255],1)
  cv.line(image_3, (0,0), (400,286), [255,255,255],1)

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


Mengalokasikan runtime

- horizontal white lines in the middle of image with certain length

```
from typing import final
image_2= cv.cvtColor(image, cv.COLOR_BGR2RGB)
image_3= cv.cvtColor(image, cv.COLOR_BGR2RGB)

cv.line(image_3, (175,215), (295,215), [255,255,255],1)

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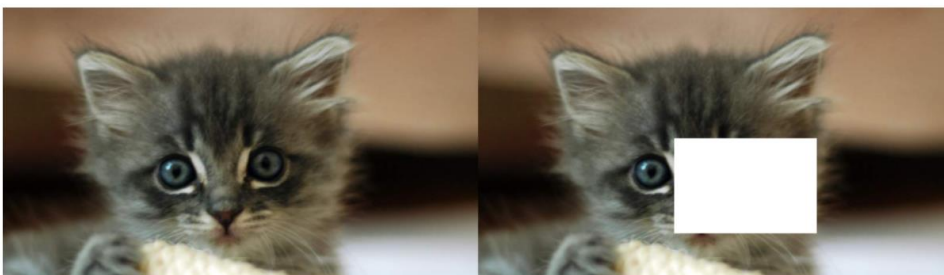

Mengalokasikan runtime

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

```
from typing import final
image_2= cv.cvtColor(image, cv.COLOR_BGR2RGB)
image_3= cv.cvtColor(image, cv.COLOR_BGR2RGB)

for y in range (110,190):
  for x in range (165,285):
    image_3[y,x]=[255,255,255]

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


Mengalokasikan runtime