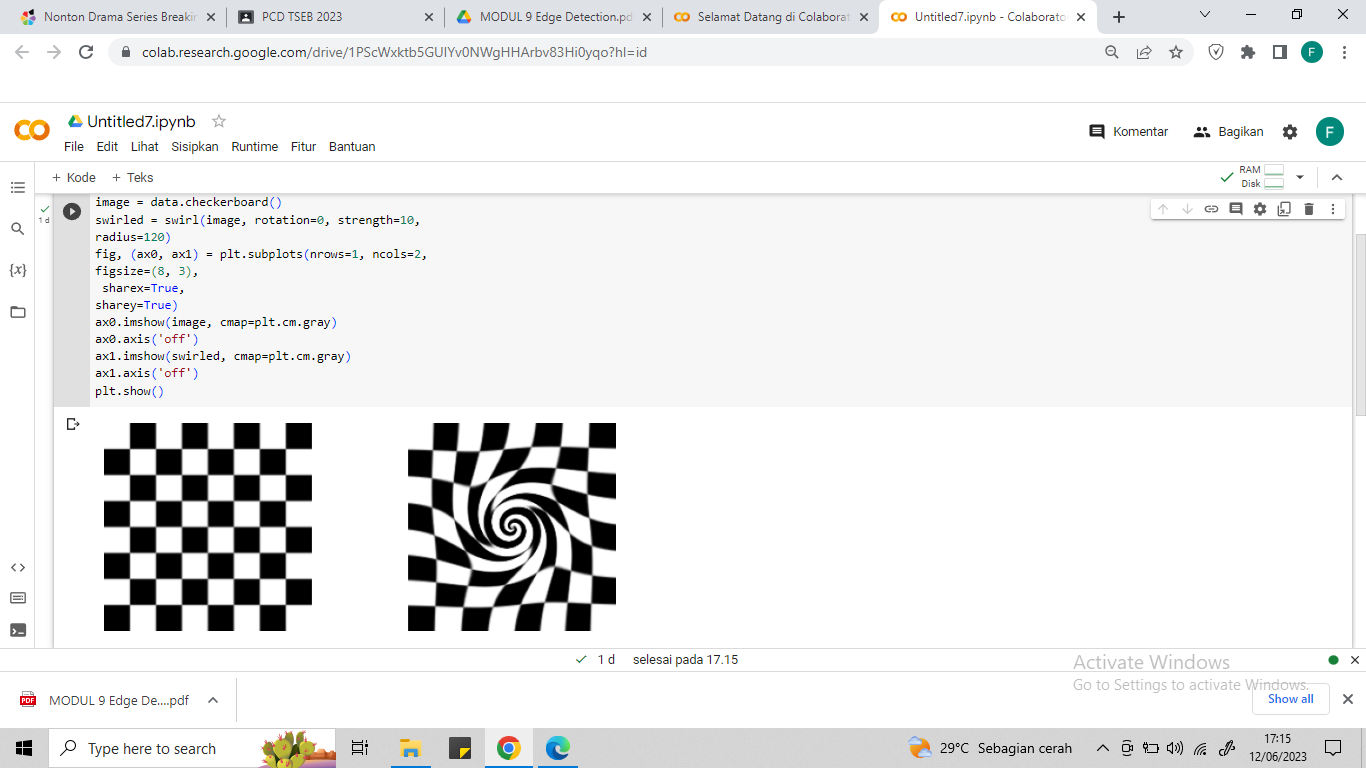
1. **TRANSFORMASI TWIRL**



import matplotlib.pyplot as plt

from skimage import data

from skimage.transform import swirl

image = data.checkerboard()

swirled = swirl(image, rotation=0, strength=10,

radius=120)

fig, (ax0, ax1) = plt.subplots(nrows=1, ncols=2,

figsize=(8, 3),

 sharex=True,

sharey=True)

ax0.imshow(image, cmap=plt.cm.gray)

ax0.axis('off')

ax1.imshow(swirled, cmap=plt.cm.gray)

ax1.axis('off')

plt.show()

1. **TRANSLATION**

import cv2 as cv

import numpy as np

import matplotlib.pyplot as plt

image = cv.imread("kucing.jpg")

h, w = image.shape[:2]

half\_height, half\_width = h//4, w//8

transition\_matrix = np.float32([[1, 0,  half\_width], [0, 1, half\_height]])

img\_transition = cv.warpAffine(image,transition\_matrix, (w, h))

plt.imshow(cv.cvtColor(img\_transition, cv.COLOR\_BGR2RGB))

plt.title("Translation") plt.show()

1. **ROTATION**

import cv2 as cv

import numpy as np

import matplotlib.pyplot as plt

image = cv.imread("kucing.jpg")

h, w = image.shape[:2]

rotation\_matrix = cv.getRotationMatrix2D((w/2,h/2), -180, 0.5)

rotated\_image = cv.warpAffine(image, rotation\_matrix, (w, h))

plt.imshow(cv.cvtColor(rotated\_image, cv.COLOR\_BGR2RGB))

plt.title("Rotation")

plt.show()

1. **INTERPOLATION**

import mahotas as mh

import numpy as np from pylab

import imshow, show

regions = np.zeros((8,8), bool)

regions[:3,:3] = 1

regions[6:,6:] = 1

labeled, nr\_objects = mh.label(regions)

imshow(labeled, interpolation='nearest')

show()

1. **SKALA INTERPOLASI MIRING**

import cv2 as cv

import numpy as np

import matplotlib.pyplot as plt

image = cv.imread("kucing.jpg")

fig, ax = plt.subplots(1, 3, figsize=(16, 8))

# image size being 0.15 times of it's original size

image\_scaled = cv.resize(image, None , fx=0.15, fy=0.15)

ax[0].imshow(cv.cvtColor(image\_scaled, cv.COLOR\_BGR2RGB))

ax[0].set\_title("Linear Interpolation Scale")

# image size being 2 times of it's original size

image\_scaled\_2 = cv.resize(image, None, fx=2, fy=2, interpolation=cv.INTER\_CUBIC)

ax[1].imshow(cv.cvtColor(image\_scaled\_2, cv.COLOR\_BGR2RGB))

ax[1].set\_title("Cubic Interpolation Scale")

# image size being 0.15 times of it's original size

image\_scaled\_3 = cv.resize(image, (200, 400), interpolation=cv.INTER\_AREA)

ax[2].imshow(cv.cvtColor(image\_scaled\_3, cv.COLOR\_BGR2RGB))

ax[2].set\_title("Skewed Interpolation Scale")