

VERSION 2.1
27 NOVEMBER 2024



[PEMROGRAMAN FUNGSIONAL]

Modul 6 - Functional Programming Implementation
for image Processing

DISUSUN OLEH:
ALVIYA LAELA LESTARI
RAFLI KHARISMA AKBAR

DIAUDIT OLEH:
FERA PUTRI AYU L., S.KOM., M.T.

LAB. INFORMATIKA
UNIVERSITAS MUHAMMADIYAH MALANG

[PEMROGRAMAN FUNGSIONAL]

PERSIAPAN MATERI

Praktikan diharapkan mempelajari terkait materi image processing dari sumber eksternal menggunakan library Pillow dan Matplotlib

TUJUAN

Sub-CPMK 7: Mahasiswa mampu mendesain program dengan teknik yang tepat untuk menyelesaikan masalah dengan menggunakan paradigma pemrograman fungsional (P6)

TARGET MODUL

1. Mampu menentukan Function yang ada dalam prosesing citra sederhana berdasarkan teknik pemrograman fungsional

PERSIAPAN SOFTWARE/APLIKASI

- Laptop/Komputer
- Windows/Linux/Mac OS
- Pycharm/Google Collab /Jupyter Notebook
- Source Code Google [Collab](#) Modul 6

Halo, tidak terasa sudah dipenghujung semester dan modul Pemrograman Fungsional. Dalam materi terakhir Pemrograman Fungsional kita akan membahas tentang Image Processing dengan python dan tentunya dengan pengimplementasian Fungsional di dalamnya. Mari kita simak penjelasan materi berikut.

1. What is Image Processing?

Menurut beberapa sumber Image processing adalah proses pengolahan sinyal dengan input berupa gambar yang kemudian diubah menjadi gambar lain sebagai outputnya menggunakan teknik tertentu. Tujuan image processing adalah untuk memperbaiki kesalahan data gambar yang terjadi selama proses transmisi dan akuisisi sinyal, serta untuk meningkatkan kualitas visual gambar agar lebih mudah diinterpretasikan oleh sistem penglihatan manusia, melalui manipulasi dan analisis gambar.

Sederhananya, Image processing adalah teknik untuk mengedit atau memodifikasi gambar digital agar menjadi lebih jelas atau memiliki kualitas yang lebih baik. Ini bisa termasuk memperbaiki warna, menghilangkan noise, memperbesar atau memperkecil gambar, dan banyak lagi. Beberapa hal dasar mengenai Image Processing meliputi:

1. **Image acquisition:** Menangkap atau mengimpor gambar melalui kamera, pemindai, dll.
2. **Preprocessing:** Mengubah gambar sebelum dianalisis (mengubah ukuran, rotasi, penghapusan noise, dll.).
3. **Feature detection:** Mengidentifikasi piksel/area yang menarik seperti tepi, sudut, atau objek.
4. **Analysis:** Mengekstraksi informasi yang bermakna dari gambar menggunakan fitur yang terdeteksi.
5. **Manipulation:** Mengubah gambar berdasarkan informasi yang telah diekstraksi (filtering, morphing, dll.).

2. Image Processing in Python

Python merupakan salah satu bahasa pemrograman yang memiliki banyak library yang mendukung Pemrosesan Citra yang mana dirangkum sebagai berikut:

- 1) **PILLOW** : [PILLOW](#) adalah versi modern dari Python Imaging Library (PIL) yang memungkinkan Anda untuk membuka, memanipulasi, dan menyimpan berbagai format gambar. Ini menyediakan fungsi dasar untuk operasi seperti pengubahan ukuran, pemotongan, rotasi, dan pengubahan warna gambar.
- 2) **Matplotlib** : [Matplotlib](#) adalah pustaka untuk membuat grafik dan visualisasi data di Python. Gambar yang dihasilkan oleh Matplotlib biasanya berupa grafik, plot, diagram, atau visualisasi data lainnya.
- 3) **OpenCV** : [OpenCV](#) (Open Source Computer Vision Library) adalah pustaka yang kuat untuk pemrosesan gambar dan visi komputer. Ini mencakup berbagai alat untuk deteksi objek, pengenalan wajah, segmentasi gambar, penghapusan noise, deteksi tepi, dan banyak lagi.
- 4) **scikit-image** : [scikit-image](#) adalah pustaka yang berfokus pada pemrosesan gambar berbasis ilmiah, dibangun di atas NumPy dan SciPy. Ini menyediakan fungsi untuk segmentasi, transformasi geometris, analisis gambar, dan deteksi fitur.
- 5) **imageio** : [imageio](#) adalah pustaka yang digunakan untuk membaca dan menulis gambar dalam berbagai format. Ini sangat berguna untuk menangani berbagai jenis file gambar, seperti GIF, JPEG, PNG, dan juga video.
- 6) **SimpleCV** : [SimpleCV](#) adalah pustaka yang dirancang untuk memudahkan pemrosesan gambar dan visi komputer. Dengan antarmuka yang sederhana, SimpleCV memudahkan pemula untuk melakukan tugas-tugas seperti pengenalan bentuk, pelacakan objek, dan pengolahan gambar dasar.
- 7) **Mahotas** : [Mahotas](#) adalah library untuk pengolahan citra berbasis perhitungan matematika dan statistik. Ini menyediakan alat untuk ekstraksi fitur, analisis morfologi, segmentasi, dan transformasi lainnya.
- 8) **Pytorch dan Tensorflow** : [Pytorch](#) dan [TensorFlow](#) adalah pustaka yang kuat untuk pembelajaran mesin dan pembelajaran mendalam (deep learning). Keduanya

sering digunakan dalam pemrosesan gambar untuk membangun dan melatih model neural network, termasuk model convolutional neural network (CNN) untuk tugas-tugas seperti pengenalan gambar dan deteksi objek.

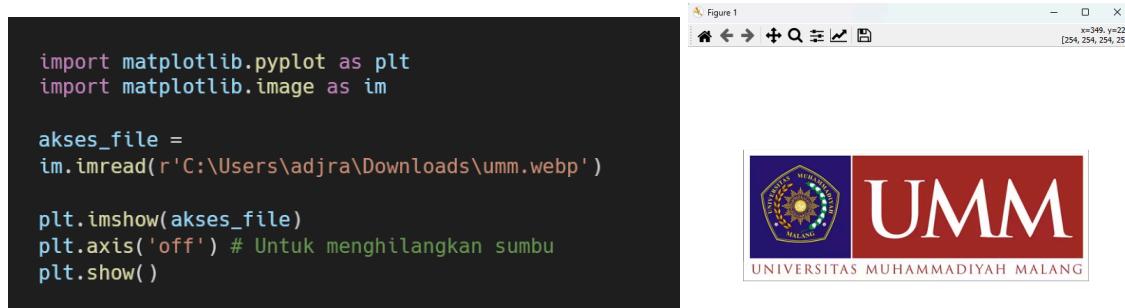
Dari banyaknya library diatas, di modul ini kita akan mempelajari image-processing sederhana menggunakan **Matplotlib** image dan **PILLOW**.

3. Image Processing with Matplotlib

Di bawah ini merupakan beberapa code dan tutorial menggunakan Matplotlib beserta function yang ada di dalamnya

3.1 Open Image File Using VSCode

Ada beberapa cara untuk mengakses file image menggunakan matplotlib salah satunya dengan penggabungan library matplotlib dan pillow. berikut contoh menggunakan 'imread' dari matplotlib untuk mengakses file dalam penyimpanan lokal kita.



```
import matplotlib.pyplot as plt
import matplotlib.image as im

akses_file =
im.imread(r'C:\Users\adjr\Downloads\umm.webp')

plt.imshow(akses_file)
plt.axis('off') # Untuk menghilangkan sumbu
plt.show()
```

contoh diatas menggunakan vscode

Dalam contoh diatas kita terlebih dahulu harus mengimport library matplotlib dan sub-library/modul apa yang ingin kita gunakan disini kita menggunakan image untuk mengakses file lokal dan pyplot untuk 'canvas' atau figur ketika kita menjalankan program. Untuk mengakses file lokal pastikan '**path**' file sudah benar jika mengalami error unicode coba menambahkan '**r**' untuk menghindari escape character.

3.2 Open Image File Using Google Collab

Ada sedikit perbedaan saat ingin mengakses image menggunakan google collab karena google collab berbasis cloud based bukan software lokal seperti visual studio code. Simak beberapa contoh berikut :

a. Local to collab

Selain mengupload file di collab seperti di modul 5, kita juga bisa menggunakan fitur upload **ketika** kita melakukan run atau meletakkan fitur upload hanya di saat code kita dijalankan seperti di bawah ini:

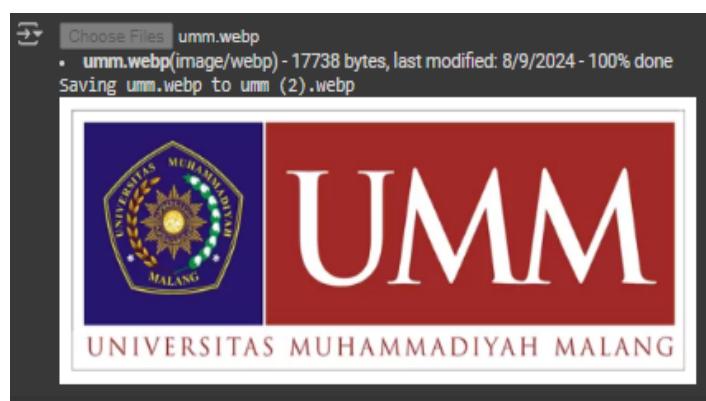
```
from google.colab import files
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

# Mengunggah gambar dari perangkat lokal ke Google Collab
uploaded = files.upload()

# Mendapatkan nama file gambar yang diunggah
uploaded_filename = next(iter(uploaded.keys()))

# Membaca gambar menggunakan Matplotlib Image
img = mpimg.imread(uploaded_filename)

plt.imshow(img)
plt.axis('off') # Tidak menampilkan sumbu koordinat
plt.show()
```



code bisa diakses di collab

b. Image from Google Drive

Ketika kita ingin mengakses google drive dalam collab kita harus melakukan mounting dan memberikan akses untuk collab.

```
# Menghubungkan Google colab dengan Google drive
from google.colab import drive
drive.mount('/content/drive')

gambar_path = '/content/drive/MyDrive/MODUL FUNGSIONAL/umm.webp' # hanya
contoh, gunakan drive pribadi untuk mengakses

img = mpimg.imread(gambar_path)

plt.imshow(img)
plt.axis('off')
plt.show()
```



Jika sudah pernah melakukan mounting, selanjutnya kita tidak perlu **menggunakan 'import drive'** lagi karena proses mounting cukup 1x saja.

3.3 Matplotlib Features

Beberapa fitur dalam pyplot yang sudah kita pelajari di modul 5 juga dapat kita implementasikan disini. Berikut beberapa fitur lain Matplotlib untuk Image Processing :

- Menambahkan judul gambar dengan title().

```
import matplotlib.pyplot as plt

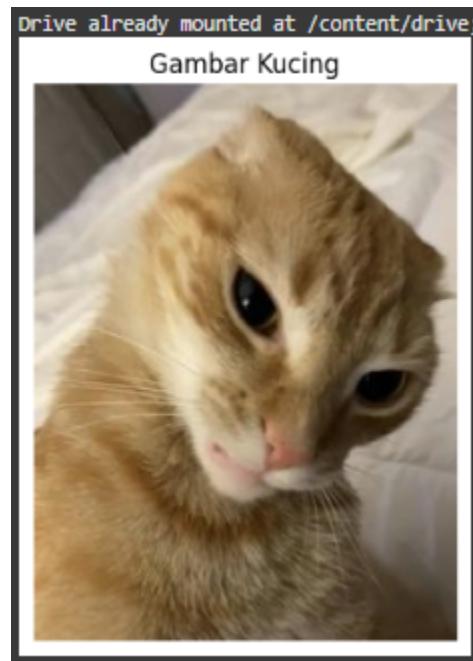
plt.imshow(image)
plt.title("Judul Gambar")
plt.show()
```

- Menampilkan atau menyembunyikan sumbu dengan axis().

```
from google.colab import drive
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

drive.mount('/content/drive')
gambar_kucing = '/content/drive/MyDrive/MODUL
FUNGSIONAL/kucing.jpg'
```

```
img = mpimg.imread(gambar_kucing)
plt.imshow(img)
plt.axis('off')
plt.title("Gambar Kucing")
plt.show()
```



- c. Menambahkan teks dan anotasi ke gambar dengan `annotate()` dan `text()`.

Contoh di bawah ini kita memberikan anotasi ke dalam gambar yang kita upload. **Anotasi** adalah memberikan informasi tambahan atau penjelasan tentang bagian tertentu dari gambar atau grafik (biasanya menggunakan panah).

```
from google.colab import drive
import matplotlib.pyplot as plt
import matplotlib.image as im
drive.mount('/content/drive')

akses_file = '/content/drive/MyDrive/MODUL FUNGSIONAL/umm.webp'
img = mpimg.imread(akses_file)
plt.imshow(img)
plt.title("Gambar dengan Anotasi")
plt.text(50, 50, 'Teks di Koordinat (50, 50)', color='yellow', fontsize=12, ha='center')

plt.annotate('Anotasi', xy=(150, 150), xytext=(200, 200),
            arrowprops=dict(facecolor='yellow', shrink=0.05),
            fontsize=12)

plt.axis('off')
plt.show()
```

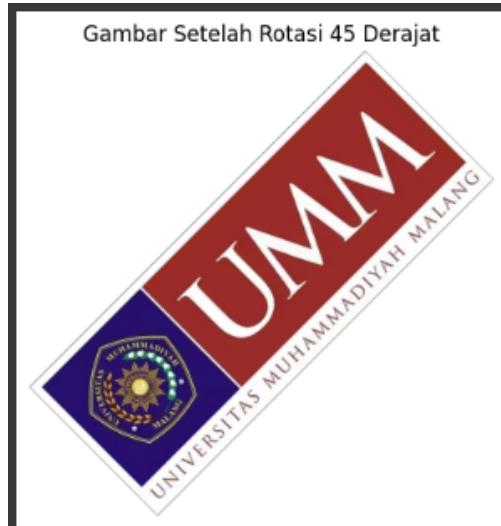


d. Transformasi Gambar menggunakan fungsi rotate

Saat menggunakan rotate maka kita akan memutar sebuah gambar sesuai dengan sumbu koordinatnya, ketika gambar di rotate() maka perputaran sumbu akan sesuai dengan **arah jarum jam**.

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import numpy as np
from scipy.ndimage import rotate

image = mpimg.imread(r'/content/drive/MyDrive/MODUL FUNGSIONAL/umm.webp')
rotated_image = rotate(image, 45, reshape=True)
plt.imshow(rotated_image)
plt.axis('off')
plt.title("Gambar Setelah Rotasi 45 Derajat")
plt.show()
```



Reshape disini berguna untuk mengatur ulang gambar, karena ketika kita rotate, seringnya gambar mengalami perubahan bentuk, reshape digunakan untuk mengantisipasinya.

e. Menyimpan gambar

```
# Menyimpan gambar
plt.imshow(image)
plt.axis('off')
plt.title("Gambar untuk Disimpan")
plt.savefig('masukkan/path/untuk/disimpan/sini')
plt.show()
```

output : Perintah plt.savefig akan menyimpan gambar ke dalam direktori yang ditentukan.

f. Filter gambar dan masih banyak lagi

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from scipy.ndimage import gaussian_filter

image = mpimg.imread('/content/drive/MyDrive/MODUL
FUNGSIONAL/umm.webp')
filtered_image = gaussian_filter(image, sigma=2)

# Menampilkan gambar yang difilter
plt.imshow(filtered_image)
plt.axis('off')
plt.title("Gambar Setelah Gaussian Filter")
plt.show()
```



Gaussian filter adalah sebuah filter yang digunakan dalam pemrosesan citra (image processing) untuk melakukan hal-hal seperti menghaluskan (smooth) atau mengurangi noise pada gambar. Filter Gaussian dengan parameter sigma=2 digunakan untuk menghaluskan (smooth) gambar, mengurangi noise, dan menghasilkan efek blur yang lebih halus. Kalian bisa mengakses contoh lain dari filter yang dimiliki oleh `scipy.ndimage` selain gaussian filter [cekidot!](#)

Selain fitur-fitur yang disebutkan diatas, matplotlib memiliki fitur yang lebih luas lagi untuk fitur Image Processing, yang bisa kalian pelajari sendiri.

3.4 Simple Image Processing using Matplotlib

```
'''fungsi kode ini memuat gambar, mengonversinya menjadi grayscale,
dan menampilkannya dalam dua bentuk, gambar asli dan gambar grayscale.
pre-cond: gambar yang dimuat harus memiliki format yang didukung oleh matplotlib (misalnya, .webp, .jpg, .png);
input gambar memiliki format RGB untuk proses konversi menjadi grayscale yang sesuai.'''
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import numpy as np

image = mpimg.imread('/content/drive/MyDrive/MODUL FUNGSIONAL/umm.webp')

plt.figure(figsize=(10,5))

plt.subplot(1, 3, 1)
plt.imshow(image)
plt.title('Gambar Awal')
plt.axis('off')

gray_image = np.dot(image[..., :3], [0.2989, 0.587, 0.114])

plt.subplot(1, 3, 2)
plt.imshow(gray_image, cmap=plt.get_cmap('gray'))
plt.title('Gambar Setelah')
plt.axis('off')
```



Kode ini memuat gambar dari file yang diberikan menggunakan `mpimg.imread()` dan menampilkannya dalam bentuk asli (RGB) pada subplot pertama, dengan ukuran plot 10×5 inci. Next, gambar tersebut dikonversi menjadi grayscale menggunakan rumus luminosity (`np.dot(image[...,:3], [0.2989, 0.587, 0.114])`) (Bisa kalian cari di internet), yang mengalikan setiap golongan warna (merah, hijau, biru)/RGB dengan koefisien yang sesuai untuk menghasilkan intensitas kecerahan yang diinginkan (coba ubah nilainya dan perhatikan bedanya). Gambar grayscale kemudian ditampilkan pada subplot kedua dengan pemetaan warna gray. Kedua gambar tersebut (gambar asli dan grayscale) ditampilkan berdampingan dalam layout 1×3 , dengan sumbu (axis) dinonaktifkan untuk tampilan yang lebih bersih.

4. Image Processing With PILLOW

PILLOW adalah library Python yang digunakan untuk memanipulasi gambar (image processing). Dengan PILLOW, kita dapat melakukan berbagai operasi pada gambar seperti membuka, menyimpan, mengedit, dan menggabungkan gambar. Library ini sangat berguna dalam pengembangan aplikasi yang melibatkan pemrosesan citra. Mari kita simak penerapan PILLOW dalam image processing :

4.1 Open Image File in Pillow

Kita sudah mengetahui bagaimana cara untuk mengakses file menggunakan library matplotlib, untuk library PILLOW tidak jauh berbeda baik menggunakan text editor dan google collab yaitu kita bisa mengakses menggunakan path local, fitur upload dan mounting ke google drive. Perbedaan hanya pada function yang dipanggil saja dan kita perlu mengimport library PIL dan memilih sub-library yang ingin kita gunakan. Mari kita simak contoh berikut :

```
from PIL import Image
import matplotlib.pyplot as plt

# Membuka gambar dari file
img = Image.open('/content/drive/MyDrive/MODUL FUNGSIONAL/umm.webp')

# Menampilkan gambar menggunakan matplotlib
plt.imshow(img)
plt.axis('off')
plt.show()
```

contoh diatas menggunakan collab dan gdrive

Perlu dicatat bahwa saat menggunakan PILLOW pada google collab, kita perlu menggunakan matplotlib sebagai 'canvas' (GUI) untuk menampilkan gambar hal ini dikarenakan function show() dalam PILLOW tidak bisa tereksekusi tanpa GUI (biasanya mengikuti environments system). Jika ingin menggunakan pure library PILLOW kita bisa menggunakan text editor seperti vscode. Berikut contoh menggunakan vscode untuk mengakses file :

```
from PIL import Image

img = Image.open(r'C:\Users\adjra\Downloads\catss.jpg')

img.show()
```

4.2 PILLOW features untuk manipulasi gambar

PILLOW memiliki fitur yang lebih lengkap untuk image processing dibanding dengan matplotlib karena matplotlib lebih diunggulkan dengan visualisasi data. Berikut beberapa contoh fitur yang ada pada PILLOW :

Gambar awal :



a. Resizing Images

Kita bisa mengatur ukuran gambar dengan pillow menggunakan resize dengan satuan **pixel**

```
from PIL import Image
import matplotlib.pyplot as plt
img = Image.open(r'/content/drive/MyDrive/MODUL FUNGSIONAL/catss.jpg')

img_resized = img.resize((200, 300))
plt.axis("off")
plt.imshow(img_resized)
plt.show()

img_resized.save('example_resized.jpg')
```



b. Cropping Images

Cropping dengan code sedikit berbeda ketika kita menggunakan fitur crop pada smartphone, crop disini menggunakan parameter yang menangkap area mana saja yang harus di crop

```
from PIL import Image
import matplotlib.pyplot as plt
img = Image.open(r'/content/drive/MyDrive/MODUL FUNGSIONAL/catss.jpg')

crop_area = (200, 100, 400, 400)
img_cropped = img.crop(crop_area)
plt.axis("off")
plt.imshow(img_cropped)
plt.show()

img_cropped.save('croppedimage.jpg')
```



(200, 100, 400 , 400) berarti posisi crop akan membentuk persegi dengan rumus (left, upper, right, lower) dari tepian gambar asli dalam satuan pixels.

c. Rotating Images

Rotating Images ada beberapa cara seperti code di bawah ini, karena biasanya saat image dirotasi bentuk images bisa berubah menyesuaikan sumbu gambar diputar, namun kita bisa memilih apakah kita ingin melakukan perubahan bentuk atau tidak.

```

from PIL import Image
import matplotlib.pyplot as plt
img = Image.open(r'/content/drive/MyDrive/MODUL FUNGSIONAL/catss.jpg')

img_rotated = img.rotate(45)
plt.axis("off")
plt.imshow(img_rotated)
plt.show()

img_rotated.save('example_rotated.jpg')

```



d. Flipping Images

Flipping image disini sama dengan membalik arah gambar dengan menggunakan fungsi transpose.

```

from PIL import Image
import matplotlib.pyplot as plt
img = Image.open(r'/content/drive/MyDrive/MODUL FUNGSIONAL/catss.jpg')

img_flipped = img.transpose(Image.FLIP_LEFT_RIGHT)
plt.axis("off")
plt.imshow(img_flipped)
plt.show()

img_flipped.save('example_flipped.jpg')

```



e. Membuat thumbnail

Fungsi thumbnail sebenarnya sama dengan kita melakukan resize dengan mempertahankan ukuran dan bentuk asli gambar dengan menggunakan fungsi thumbnail, fungsi thumbnail tidak memiliki default value oleh sebabnya kita harus inialisasi terlebih dahulu.

```

img.thumbnail((128, 128))
img.save('example_thumbnail.jpg')

```



f. Merging Images/Overlay Images

Merging atau overlay sama halnya dengan yang kita lakukan untuk mengedit foto, kita perlu 2 gambar awal untuk digabungkan menjadi sebuah overlay.

```

from PIL import Image
import matplotlib.pyplot as plt

img = Image.open(r'/content/drive/MyDrive/MODUL FUNGSIONAL/catss.jpg')
img2 = Image.open(r'/content/drive/MyDrive/MODUL FUNGSIONAL/catssredd.jpg')
img2 = img2.resize(img.size)

if img.mode != img2.mode:
    img2 = img2.convert(img.mode)

merged_image = Image.blend(img, img2, alpha=0.5)
merged_image.save('example_merged.jpg')

plt.imshow(merged_image)
plt.axis('off')
plt.show()

```



g. Images Enhancer

Image enhancer dalam pillow memiliki beberapa fitur seperti mengatur brightness, colour, contrast, sharpness. Untuk contoh lebih lanjut bisa kamu pelajari di [ImageEnhance](#).

```

from PIL import Image, ImageEnhance
import matplotlib.pyplot as plt

img = Image.open(r'/content/drive/MyDrive/MODUL FUNGSIONAL/catss.jpg')
# Meningkatkan kecerahan
enhancer_brightness = ImageEnhance.Brightness(img)
img_brightness = enhancer_brightness.enhance(1.5) # kelipatan brightness

# Meningkatkan kontras
enhancer_contrast = ImageEnhance.Contrast(img_brightness) # Memperbaiki
nama variabel
img_contrast = enhancer_contrast.enhance(1.5) # kelipatan kontras

plt.imshow(img_contrast)
plt.axis('off')
plt.show()
img_contrast.save('example_enhanced.jpg')

```



h. Filtering Images

Dalam PILLOW terdapat banyak function untuk melakukan [filtering](#), seperti yang tercakup di bawah ini :

- BLUR

- CONTOUR
- DETAIL
- EDGE_ENHANCE
- EDGE_ENHANCE_MORE
- EMBOS
- FIND_EDGES
- SHARPEN
- SMOOTH
- SMOOTH_MORE

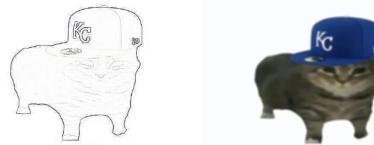
```
from PIL import ImageFilter, Image
import matplotlib.pyplot as plt

img = Image.open(r'/content/drive/MyDrive/MODUL FUNGSIONAL/catss.jpg')

# Terapkan efek filter
img_blur = img.filter(ImageFilter.BLUR)
img_contour = img.filter(ImageFilter.CONTOUR)

# Menyimpan gambar hasil filter
img_blur.save('example.blur.jpg')
img_contour.save('/example_contour.jpg')

# Tampilkan gambar before and after (kiri: asli, kanan: filter)
plt.subplot(1, 2, 1)
plt.imshow(img)
plt.axis('off')
plt.subplot(1, 2, 2)
plt.imshow(img_contour)
plt.axis('off')
plt.show()
```



Seperti halnya library pada umumnya banyak fitur yang mungkin tidak akan lengkap dijelaskan disini seperti palette, path, sequence dan masih banyak lagi kalian bisa mengexplore sendiri pada tautan [berikut](#).

4.3 Simple Image Processing with PILLOW



```
from PIL import Image, ImageEnhance, ImageFilter
import matplotlib.pyplot as plt

img = Image.open(r'/content/drive/MyDrive/MODUL
FUNGSIONAL/kucing_berdasi.jpg')
enhancer = ImageEnhance.Brightness(img)
img_enhanced = enhancer.enhance(3)
img_blurred = img_enhanced.filter(ImageFilter.GaussianBlur(radius=3))
img_gray = img_blurred.convert('L')
img_gray.save('kucing.jpg')
plt.imshow(img_gray, cmap='gray')
plt.axis('off')
plt.show()
```

Output :



Penjelasan :

- 1) Kita menggunakan variabel enhancer untuk memastikan bahwa ImageEnhance berfokus pada adjustment Brightness.
- 2) Selanjutnya melakukan peningkatan 3x lipat brightness dari semula
- 3) Terakhir kita menambahkan gaussian filter untuk gambar

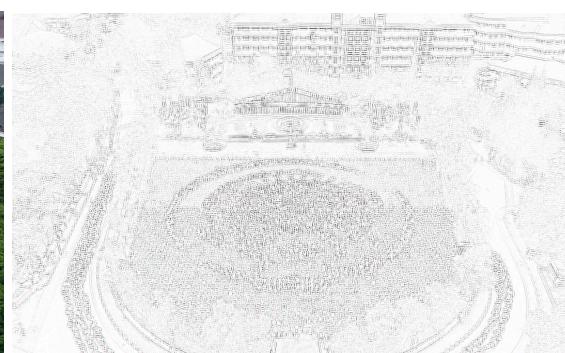
CODELAB

CODELAB 1

Before :



After :



Lakukan simple image processing dengan hasil seperti diatas (manfaatkan materi subplot modul 5). Kalian bisa melengkapi code berikut sebagai panduan untuk mengerjakan.

```
from PIL import Image,  
  
img = Image.open()  
image_after = img.filter(ImageFilter.)  
image_after2 = .filter(ImageFilter.CONTOUR)  
.show()
```

**) Kalian bisa menggunakan materi filtering images dalam sub materi pillow hasil gambar menggunakan 2 mode filter*

CODELAB 2

1. Lakukanlah percobaan program dari implementasi sub-materi [Matplotlib Features](#) diantaranya
 - a. Membuat judul
 - b. Menyembunyikan sumbu Axis
 - c. Menambahkan Teks dan anotasi
 - d. Rotate gambar
 - e. Menyimpan gambar
 - f. Melakukan filtering
2. Gunakan gambar bebas yang ada di pc kalian

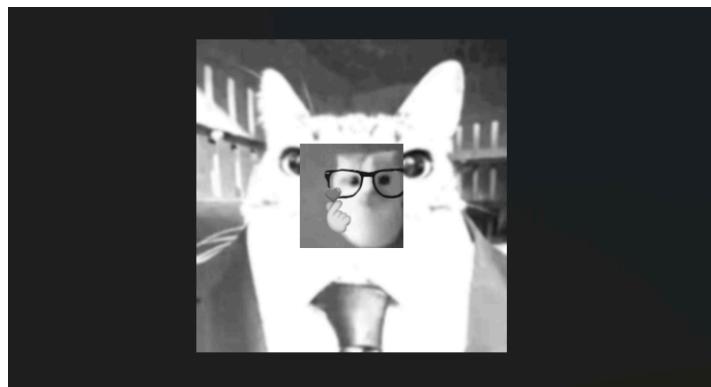
CODELAB 3

1. Pilihlah 2 gambar dari komputer anda untuk membuat sebuah overlay image.
2. Buka gambar pertama sebagai latar belakang dan gambar kedua yang akan disisipkan (overlay)
3. Untuk gambar kedua kalian bisa melakukan resize() terlebih dahulu (optional) agar gambar 2 yang menjadi overlay tidak lebih besar dari background image, kalian bisa membuat skala (100,100) pixel.
4. Sisipkan gambar overlay ke dalam gambar background menggunakan fungsi paste()
5. Simpan gambar hasil edit
6. Tampilkan hasil edit kepada asisten

*) hint : gunakanlah fungsi overlay/merging images setelah itu gunakan fungsi resize dan simpan gambar

```
# TODO 1 : Buka gambar
background_image =
overlay_image =
# TODO 2 : Konversi overlay ke mode RGB (menghilangkan alpha channel)
overlay_image = overlay_image.convert()
# TODO 3 : (Optional) Perkecil ukuran gambar overlay menggunakan
resize()
# TODO 4 : Sisipkan gambar overlay ke dalam gambar background
background_image.paste(overlay_image, (x_center, y_center))
# TODO 5 : Simpan gambar hasil edit
# TODO 6 : Tampilkan{
```

Contoh output (hanya contoh) :



TUGAS

TUGAS 1

Gambar Awal



Gambar Kedua



Lakukanlah eksperimen mandiri terkait image processing dari gambar awal sehingga

menghasilkan hasil image seperti di bawah ini. Semakin mirip dan tepat terkait proses yang kalian buat maka nilai semakin baik.

TUGAS 2

Buatlah program dari soal cerita di bawah ini!

Pak Ronaldo sedang mencoba untuk membuka toko onlinenya, beliau ingin mengupload foto-foto produk di etalase online miliknya namun Pak Ronaldo merasa bahwa foto yang diupload kurang sesuai dengan kehendaknya. Beliau ingin menyunting foto produknya dengan poin-poin di bawah ini :

1. Kecerahan gambar terlalu gelap, beliau ingin meningkatkannya sebanyak 30%.
2. Ukuran gambar-gambarnya tidak sesuai satu dengan lainnya beliau ingin ukurannya 300×300 pixel agar konsisten
3. Beliau ingin memberikan border (warna bebas) dengan lebar 10 pixel agar lebih rapi
4. Beliau ingin foto produknya lebih tajam dari sebelumnya.
5. Beliau ingin mengubah warna gambar menjadi grayscale jika sewaktu-waktu produknya habis.
6. Tambahkan watermark merk toko beliau di ujung kanan gambar.
7. Beliau ingin gambar yang sudah diedit disimpan dalam perangkat pribadi Pak Ronaldo
8. Beliau ingin ada perbandingan antara gambar asli dan gambar hasil manipulasi menggunakan subplot pada matplotlib.

Note :

1. Untuk gambar katalog produk, nama toko, dan sejenisnya dibebaskan
2. Gunakan matplotlib sebagai kanvas, dan sublibrary dari PILLOW untuk melakukan image processing.
3. Bangun program menggunakan paradigma fungsional sehingga dapat dimanfaatkan untuk memproses berapapun gambar yang dimiliki Pak Ronaldo. Sebagai contoh, kalian bisa membuat fungsi editing, watermarking, dan showing berdasarkan kriteria yang telah ditentukan.

KRITERIA & DETAIL PENILAIAN

KETERANGAN		POIN	PROGRAM IDENTIK
Codelab	Codelab 1, 2, & 3 masing-masing 5 poin	15	15
Tugas 1	Pemahaman dan Ketepatan program	25	21
	Code/Kelengkapan fitur	15	0
Tugas 2	Pemahaman dan Ketepatan program	25	21
	Code/Kelengkapan fitur	20	0
Total (Advance)		100	57

*)Note: Program Identik berarti program sama persis dengan praktikan lain sehingga yang dinilai hanya pemahaman terhadap materi (code tidak mendapat bobot nilai sama sekali).

**)Poin diatas merupakan poin maksimal yang bisa diperoleh. Asisten bisa memberikan nilai dibawah itu jika dirasa praktikan tidak maksimal saat demo (kurangnya pemahaman tentang apa yang di demokan).