

VERSION 2.1  
AGUSTUS, 2023



# [PRAKTIKUM PEMROG. FUNGSIONAL]

## MODUL 6 – Implementasi Pemrograman Fungsional dalam studi kasus Pemrosesan Gambar

DISUSUN OLEH :  
Fildzah Lathifah  
Hania Pratiwi Ningrum

DIAUDIT OLEH  
Fera Putri Ayu L., S.Kom., M.T.

PRESENTED BY: TIM LAB-IT UNIVERSITAS  
MUHAMMADIYAH MALANG

## [PRAKTIKUM PEMROG. FUNGSIONAL]

---

### PERSIAPAN MATERI

Praktikan diharapkan juga mempelajari dari sumber eksternal mengenai library PILLOW dan Matplotlib.

---

### TUJUAN PRAKTIKUM

1. Praktikan diharap dapat mengimplementasikan pemrograman fungsional dalam pemrosesan citra
  2. Praktikan mampu menentukan Function yang ada dalam pemrosesan citra berdasarkan teknik pemrograman fungsional
- 

### TARGET MODUL

Penguasaan materi:

1. Pemrosesan citra sederhana dengan library Matplotlib dan PILLOW
- 

### PERSIAPAN SOFTWARE/APLIKASI

- Komputer/Laptop
  - Sistem operasi Windows/Linux/Mac OS/Android
- 

### MATERI POKOK

Pada pemrograman fungsional, modul juga dapat diakses melalui google collab agar lebih interaktif :

[Modul 6](#)

Wahh tidak terasa kita sudah sampai pada modul terakhir, pada modul 6 kali ini kita akan mengenal sebuah library bernama PILLOW dan Matplotlib Image. Dua library ini banyak dipakai untuk pemrosesan citra. Sebelum memasuki materi lebih dalam, mari kita cari tahu terlebih dahulu apa itu pemrosesan gambar/citra.

## 1. Pemrosesan Gambar dan Library-nya

Pengolahan citra adalah metode untuk melakukan beberapa operasi pada gambar, untuk mendapatkan gambar yang disempurnakan atau untuk mengekstrak beberapa informasi berguna darinya. Pada pengolahan citra, gambar dijadikan input, dan hasilnya dapat berupa gambar yang telah diolah atau fitur-fitur yang dapat diambil dari gambar.

Pengolahan citra sebenarnya tidak hanya mengenai aspek estetika atau keindahan saja, tetapi juga tentang menyesuaikan persepsi manusia dengan perangkat citra digital. Ini penting karena sistem penglihatan manusia dan komputer memiliki perbedaan dalam cara kerjanya. Komputer memiliki pembatasan dalam hal noise dan bandwidth.

Terdapat beberapa library populer dalam bahasa pemrograman Python yang dapat digunakan untuk melakukan pengolahan citra. Berikut beberapa di antaranya:

1. **PILLOW (PIL Fork):** PILLOW adalah library yang umum digunakan untuk operasi dasar pengolahan citra seperti membuka, menyimpan, dan memanipulasi gambar. Ini adalah pengganti yang lebih modern dari Python Imaging Library (PIL).
2. **OpenCV:** OpenCV (Open Source Computer Vision Library) adalah library sumber terbuka yang sangat kuat untuk pengolahan citra dan komputer. Ini memiliki berbagai fungsi untuk analisis citra, deteksi objek, pengenalan wajah, serta dukungan untuk pengolahan video.
3. **scikit-image:** Ini adalah modul dalam library scikit-learn yang didedikasikan untuk pengolahan citra. Ia menawarkan berbagai fungsi untuk segmentasi gambar, analisis fitur, ekstraksi fitur, dan transformasi.
4. **imageio:** imageio adalah library yang berguna untuk membaca dan menulis berbagai jenis format gambar. Ini sangat berguna ketika Anda perlu membaca atau menyimpan gambar dalam format yang tidak secara langsung didukung oleh PILLOW atau OpenCV.
5. **SimpleCV:** SimpleCV adalah library yang lebih tingkat tinggi yang dibangun di atas OpenCV. Ini dirancang untuk mempermudah pemula dalam pengolahan citra

dengan menyediakan antarmuka yang lebih mudah digunakan.

6. **Mahotas**: Mahotas adalah library untuk pengolahan citra berbasis perhitungan matematika dan statistik. Ini menyediakan alat untuk ekstraksi fitur, analisis morfologi, segmentasi, dan transformasi lainnya.
7. **PyTorch dan TensorFlow**: Meskipun secara utama dikenal sebagai library untuk deep learning, PyTorch dan TensorFlow juga memiliki kemampuan untuk melakukan manipulasi citra dan pemrosesan sinyal, terutama dalam konteks jaringan saraf tiruan.

Semua library ini memiliki kelebihan dan kekurangan masing-masing, dan pilihan tergantung pada kebutuhan proyek dan tingkat kenyamanan dengan antarmuka dan fitur-fitur yang ditawarkan oleh masing-masing library. Namun kali ini kita akan fokus untuk mempelajari dua library terlebih dahulu yaitu PILLOW dan Matplotlib Image.

## 2. Cara membuka gambar dengan library Matplotlib.image

Sebelum masuk pada materi library, mari kita belajar cara membuka gambar terlebih dahulu. Jika menggunakan IDE, gunakan kode seperti dibawah ini:

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

# Membaca dan menampilkan gambar
img = mpimg.imread('direktori/gambar.png') #Sesuaikan dengan direktori
letak gambar tersimpan
plt.imshow(img)
plt.axis('off') # Tidak menampilkan sumbu koordinat
plt.show()
```

Taraaa, maka hasilnya akan seperti gambar dibawah ini. Jangan lupa untuk memastikan direktori sudah benar ya

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

# Membaca dan menampilkan gambar
img = mpimg.imread('C:\dev\Modul 6\logo umm.jpg') #Sesuaikan dengan direktori letak gambar tersimpan
plt.imshow(img)
plt.axis('off') # Tidak menampilkan sumbu koordinat
plt.show()

✓ 3.6s
```



Untuk mencoba latihan diatas, salin kode ke dalam IDE yang kalian punya, jangan lupa ubah direktori sesuai penyimpanan pada perangkat masing masing ya. lalu tunjukkan hasilnya kepada asisten

Eitsss, Lalu bagaimana caranya jika kita menggunakan Google collab?? Ikuti cara dibawah ini yaaa

### a. Membuka Gambar dari Perangkat Lokal

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

# 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)

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

no files selected

Please rerun this cell to enable.

Saving logo umm.jpg to logo umm.jpg

Upload widget is only available when the cell has been executed in the current browser session.



Jalankan ulang kode diatas, agar mengetahui proses yang berlangsung

## b. Membuka Gambar dari Google Drive

Untuk dapat mengakses GDrive, kita perlu menghubungkan drive dengan collab terlebih dahulu. Kalian bisa menggunakan library drive untuk memanggil fungsi mount() seperti berikut:

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

Saat pertama kali menjalankan kode diatas, collab akan meminta izin untuk mengakses drive, dan kalian perlu login ke dalam drive agar bisa terhubung. Setelah berhasil, maka kita dapat mengakses file apapun yang ada dalam drive kita. Untuk kasus kali ini, kita hanya akan mengolah file gambar, jadi pastikan dalam drive kalian sudah tersedia file gambar yang akan kita olah. Jika belum, kalian bisa meng-upload-nya terlebih dahulu ke dalam drive.

Selanjutnya, kita bisa memproses imread() dengan menyesuaikan kan gambar\_path sesuai dengan tempat gambar tersimpan pada drive.

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

# Path ke gambar di Google Drive
gambar_path = '/content/drive/MyDrive/UMM/logo umm.jpg'
```

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

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



Jalankan lagi kode diatas untuk latihan, jangan lupa hubungkan drive dengan collab terlebih dahulu ya, lalu sesuaikan gambar\_path sesuai dengan tempat gambar tersimpan pada drive kamu. Cara melihat apakah collab sudah terhubung dengan drive bisa klik gambar folder pada kanan atas tampilan google colab

### 3. Library 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.

Berikut adalah beberapa fungsi penting dalam pengolahan citra menggunakan library PILLOW:

#### Membuka dan Menyimpan Gambar

- `Image.open(path)`: Membuka gambar dari file.
- `image.save(path)`: Menyimpan gambar ke file.

```
from PIL import Image

# Membuka gambar dari file
img = Image.open('/content/logo python (1).png')

# Menyimpan gambar ke file
img.save('/content/gambar_baru.png')
```

## Manipulasi Ukuran dan Skala

- `image.resize(size)`: Mengubah ukuran gambar.
- `image.thumbnail(size)`: Membuat salinan gambar dengan ukuran maksimum.

```
# pastikan kalian telah melakukan open image seperti pada kode sebelumnya
# untuk melanjutkan kode berikut

# Mengubah ukuran gambar
new_size = (800, 600)
resized_image = img.resize(new_size)

# Membuat salinan gambar dengan ukuran maksimum
max_size = (400, 400)
img.thumbnail(max_size)
```

## Pemotongan dan Rotasi

- `image.crop(box)`: Memotong gambar berdasarkan koordinat kotak.
- `image.rotate(angle)`: Memutar gambar sebesar sudut tertentu.

```
# pastikan kalian telah melakukan open image seperti pada kode sebelumnya
# untuk melanjutkan kode berikut

# Memotong gambar berdasarkan koordinat kotak (left, upper, right, lower)
box = (100, 100, 400, 300)
cropped_image = img.crop(box)

# Memutar gambar sebesar sudut tertentu (misalnya 45 derajat)
rotated_image = img.rotate(45)
```

## Penerapan Efek

- `ImageFilter`: Modul untuk mengaplikasikan efek seperti blur, sharpen, dan



lainnya.

```
# pastikan kalian telah melakukan open image seperti pada kode sebelumnya
# untuk melanjutkan kode berikut

# Menggunakan filter blur
blurred_image = img.filter(ImageFilter.BLUR)

# Menggunakan filter sharpen
sharpened_image = img.filter(ImageFilter.SHARPEN)
```

### Perubahan Warna

- `image.convert(mode)`: Mengubah mode warna (RGB, grayscale, dll).
- `image.filter(filter)`: Mengubah tampilan gambar dengan filter warna.

```
# pastikan kalian telah melakukan open image seperti pada kode sebelumnya
# untuk melanjutkan kode berikut

# Mengubah mode warna gambar ke grayscale
grayscale_image = img.convert('L')

# Mengubah mode warna gambar ke RGB ditambah filter warna sepia
sepia_image = img.convert('RGB', matrix=[(0.272, 0.534, 0.131),
                                           (0.349, 0.686, 0.168),
                                           (0.393, 0.769, 0.189)])

# Mengubah tampilan gambar menggunakan filter Gaussian Blur
blurred_image = img.filter(ImageFilter.GaussianBlur(radius=2))
```

## 4. Library Matplotlib Image

Matplotlib Image adalah bagian dari Matplotlib yang fokus pada visualisasi dan manipulasi gambar. Ini memungkinkan Anda untuk dengan mudah menampilkan gambar dalam berbagai format, menerapkan anotasi, dan melaksanakan tugas-tugas pemrosesan citra dasar.

### Beberapa fitur utama Matplotlib Image:

- Menampilkan gambar dengan `imshow()`, yang mendukung berbagai format gambar.

- Menambahkan judul gambar dengan title().
- Menampilkan atau menyembunyikan sumbu dengan axis().
- Menambahkan teks dan anotasi ke gambar dengan annotate() dan text().
- Plotting histogram dari citra menggunakan hist().

Contoh penerapan dari beberapa fungsi diatas adalah sebagai berikut:

```
import matplotlib.pyplot as plt

#1. Menampilkan gambar
plt.imshow(image)
plt.title("Judul Gambar")
plt.show()

#2. Anotasi dan Teks
plt.annotate("Teks Anotasi", xy=(x, y), xytext=(x_text, y_text),
arrowprops=dict(arrowstyle="->"))
plt.text(x, y, "Teks Tertentu")

#3. Plotting Histogram
plt.hist(image.ravel(), bins=256, range=(0, 256), density=True,
color='gray', alpha=0.6)
```

Contoh penerapan pemrosesan citra sederhana:

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

drive.mount('/content/drive') #jangan lupa hubungkan drive dengan collab
ya

# Membuka gambar dengan PILLOW
image_path = "/content/drive/MyDrive/UMM/logo umm.jpg" # Pastikan path
sesuai dengan nama file yang sebenarnya
image = Image.open(image_path)

# Memastikan bahwa gambar memiliki format RGB
if image.mode != "RGB":
    image = image.convert("RGB")

resized_image = image.resize((300, 200))
blurred_image = resized_image.filter(ImageFilter.GaussianBlur(radius=2))
```

```
# Menampilkan gambar dengan Matplotlib Image
plt.imshow(blurred_image)
plt.title("\nGambar yang Diblur")
plt.axis("off")
plt.show()
```

Gambar yang Diblur



Coba jalankan kembali kode diatas dengan gambar sesuai yang kamu inginkan

---

## LATIHAN PRAKTIKUM

### KEGIATAN PERCOBAAN 1 (mengubah warna gambar dan menambahkan teks)

Pilih/tentukan satu gambar yang akan kamu edit

1. Buka gambar menggunakan Pillow.
2. Ubah gambar menjadi hitam-putih (grayscale).
3. Tambahkan teks berisikan nama dan NIM kalian di tengah gambar dengan font Arial dan ukuran 24. Sebelumnya jangan lupa unduh terlebih dahulu font yang kamu gunakan ya, lalu panggil sesuai dengan direktori tempat kamu menyimpan hasil unduhan.
4. Simpan gambar hasil edit dengan nama "percobaan\_satu.jpg".
5. Tampilkan hasil pengolahan gambar kalian.

```
#Lengkapi kode dibawah untuk menjawab soal diatas ya

# TODO 0 : Import library lain yang dibutuhkan
from PIL import ImageDraw, ImageFont
```

```

# TODO 1: Lakukan load image pada variabel berikut
# hint: kalian bisa gunakan fungsi open()
gambarku =

# TODO 2: Ubah gambar menjadi hitam-putih
# hint: kalian bisa gunakan fungsi convert()
gambarBW = gambarku.convert(" ")

# TODO 3: Tambahkan text sesuai kriteria.
draw = ImageDraw.Draw(gambarBW)
font = ImageFont.truetype(direktoriFont, size)
text = ""
text_width, text_height = draw.textsize(text, font)
text_x = (gambarku.width - text_width) // 2
text_y = 20
draw.text()

# TODO 4: Simpan gambar hasil edit menggunakan fungsi save()

# TODO 5: Tampilkan hasil akhir gambar

```

## KEGIATAN PERCOBAAN 2 (menyisipkan gambar ke gambar lainnya)

Anda memiliki dua gambar:

- gambar latar belakang (background image) berupa pemandangan/environment dan
- gambar objek yang ingin disisipkan (overlay image).

Anda ingin menyisipkan gambar kedua ke dalam gambar pertama menggunakan library Pillow:

1. Buka gambar latar belakang (background) dan gambar yang ingin disisipkan (overlay)
2. Konversi overlay image ke mode RGB (menghilangkan alpha channel)
3. (Optional) Perkecil ukuran gambar overlay menggunakan method `resize()`
4. Sisipkan gambar overlay ke dalam gambar background menggunakan method `paste()`
5. Simpan gambar hasil edit
6. Tampilkan hasil edit gambar

```

#Lengkapi kode dibawah ini untuk menjawab soal diatas ya

# TODO 0 : Import beberapa library lain yang dibutuhkan

# TODO 1 : Buka gambar latar belakang (background) dan gambar yang ingin
disisipkan (overlay)

```

```

background_image =
overlay_image =

# TODO 2 : Konversi overlay image ke mode RGB (menghilangkan alpha
channel)
overlay_image = overlay_image.convert()

# TODO 3 : (optional) Perkecil ukuran gambar overlay menggunakan method
resize()

# Tentukan/Hitung koordinat tengah untuk menempatkan overlay
x_center =
y_center =

# TODO 4 : Sisipkan gambar overlay ke dalam gambar background menggunakan
method paste()
background_image.paste(overlay_image, (x_center, y_center))

# TODO 5 : Simpan gambar hasil edit

# TODO 6 : Tampilkan

```

### KEGIATAN PERCOBAAN 3 (Mengubah Tingkat Kecerahan dan Kontras Gambar)

Anda memiliki gambar berikut dan ingin mengubah tingkat kecerahan dan kontrasnya menggunakan library Pillow:

1. Buka gambar menggunakan Pillow.
2. Ubah tingkat kecerahan menjadi 1.5 kali lipat dan tingkat kontras menjadi 1.2 kali lipat.
3. Simpan gambar hasil edit dengan nama "brightness\_contrast\_image.jpg".
4. Tampilkan hasilnya

```

from PIL import Image, ImageEnhance

# TODO 1: Buka gambar dengan Pillow

# TODO 2: Ubah tingkat kecerahan (brightness) dan kontras (contrast)
enhancer = ImageEnhance.Brightness(image)
brightened_image = enhancer.enhance(1.5)

enhancer = ImageEnhance.Contrast(brightened_image)
final_image = enhancer.enhance(1.2)

# TODO 3: Simpan gambar hasil edit

# TODO 4: Tampilkan

```

```

#Tulis kembali dan lengkapi todo yang telah diberikan
#Jangan lupa untuk menggabungkan drive terlebih dahulu ya untuk memuat
gambar

```

#Buat nilai contrast dan brightness baru yang berbeda dan tampilkan hasilnya

## TUGAS PRAKTIKUM

Telah disediakan dua buah gambar background dan logo dan Anda diminta untuk menyisipkan gambar kedua (logo) ke dalam gambar pertama menggunakan library Pillow dan melakukan beberapa perubahan untuk gambar sebagai berikut:

Gambar latar belakang: *(unduh gambar pada google collab yang sudah disediakan)*

Gambar yang ingin disisipkan: *(unduh gambar pada google collab yang sudah disediakan)*

1. Buka kedua gambar menggunakan Pillow.
2. Ubah gambar background menjadi hitam-putih (grayscale), berotasi sebesar 30 derajat, dan blur
3. Ubah tingkat kecerahan gambar overlay menjadi 1.x kali lipat dan tingkat kontras menjadi 1.y kali lipat. Dimana nilai x dan y mengacu pada 2 digit NIM akhir kalian dan resize sesuai kebutuhan.
4. Sisipkan gambar overlay ke dalam gambar background sehingga terlihat seperti stiker pada gambar latar belakang.
5. Tambahkan teks "Informatika JOSSS!" pada gambar overlay dengan font Arial dan ukuran 24.
6. Simpan gambar hasil edit dengan nama "tugas\_praktikum\_enam.jpg".

Tampilan gambar yang akan diedit:



## KRITERIA & DETAIL PENILAIAN TUGAS PRAKTIKUM

Kriteria	Bobot	Program identik*
1. Praktikan berhasil melakukan <b>processing background image</b> sesuai kriteria yang diminta	20	10*
2. Praktikan berhasil melakukan <b>processing overlay image</b> (logo) sesuai kriteria yang diminta	20	10*
3. Praktikan berhasil <b>menyisipkan gambar</b> yang telah diproses sebelumnya	20	10*
4. Praktikan berhasil <b>menambahkan text dan menyimpan</b> gambar hasil pengolahan	20	10*
5. <b>Menjelaskan dan menjawab pertanyaan</b> asisten saat demo dengan baik dan benar	20	20*
Total Nilai (maksimal)	100	60*

\*) Jika program identik dengan praktikan lain. Maka akan ada pengurangan nilai pada kedua praktikan