

**LAPORAN UAS
MATA KULIAH
PENGOLAHAN GAMBAR DAN FOTOGRAFI**

Topik
Sagmentasi Citra



PENYUSUN LAPORAN



Nama Mahasiswa	NIM	Kelas
SAFIRA ADILA	062340833201	1 MIN

**PROGRAM STUDI MANAJEMEN INFORMATIKA JURUSAN
MANAJEMEN INFORMATIKA
POLITEKNIK NEGERI SRIWIJAYA
2023**

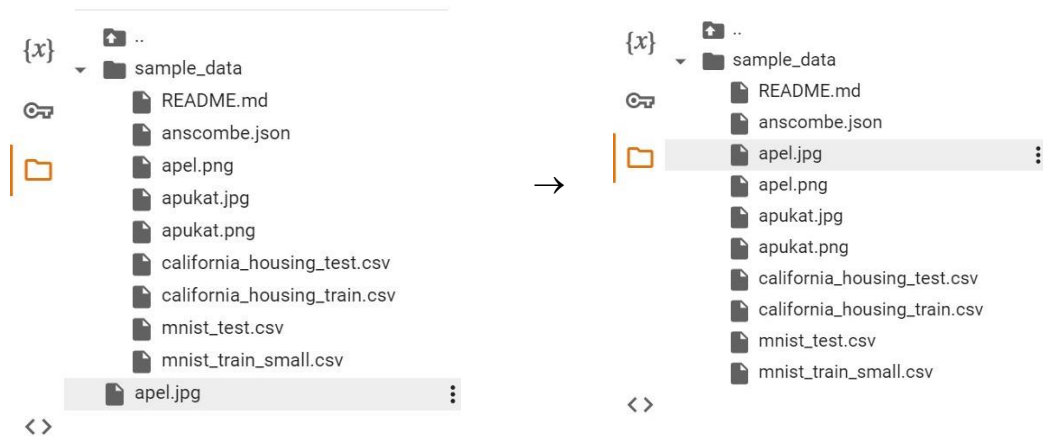
LANGKAH - LANGKAH :

1. Install library remove background terlebih dahulu untuk segmentasi citra, kode nya seperti berikut :

```
pip install rembg[cli,gpu]
```

```
Requirement already satisfied: rembg[cli,gpu] in /usr/local/lib/python3.10/dist-packages (2.0.53)
Requirement already satisfied: jsonschema in /usr/local/lib/python3.10/dist-packages (from rembg[cli,gpu]) (4.19.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from rembg[cli,gpu]) (1.23.5)
Requirement already satisfied: onnxruntime in /usr/local/lib/python3.10/dist-packages (from rembg[cli,gpu]) (1.16.3)
Requirement already satisfied: opencv-python-headless in /usr/local/lib/python3.10/dist-packages (from rembg[cli,gpu]) (4.9.0.80)
Requirement already satisfied: pillow in /usr/local/lib/python3.10/dist-packages (from rembg[cli,gpu]) (9.4.0)
Requirement already satisfied: pooch in /usr/local/lib/python3.10/dist-packages (from rembg[cli,gpu]) (1.8.0)
Requirement already satisfied: pymatting in /usr/local/lib/python3.10/dist-packages (from rembg[cli,gpu]) (1.1.12)
Requirement already satisfied: scikit-image in /usr/local/lib/python3.10/dist-packages (from rembg[cli,gpu]) (0.19.3)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from rembg[cli,gpu]) (1.11.4)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from rembg[cli,gpu]) (4.66.1)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-packages (from rembg[cli,gpu]) (3.9.1)
Requirement already satisfied: asyncer in /usr/local/lib/python3.10/dist-packages (from rembg[cli,gpu]) (0.0.2)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from rembg[cli,gpu]) (8.1.7)
Requirement already satisfied: fastapi in /usr/local/lib/python3.10/dist-packages (from rembg[cli,gpu]) (0.108.0)
Requirement already satisfied: filetype in /usr/local/lib/python3.10/dist-packages (from rembg[cli,gpu]) (1.2.0)
Requirement already satisfied: gradio in /usr/local/lib/python3.10/dist-packages (from rembg[cli,gpu]) (4.13.0)
Requirement already satisfied: python-multipart in /usr/local/lib/python3.10/dist-packages (from rembg[cli,gpu]) (0.0.6)
Requirement already satisfied: uvicorn in /usr/local/lib/python3.10/dist-packages (from rembg[cli,gpu]) (0.25.0)
Requirement already satisfied: watchdog in /usr/local/lib/python3.10/dist-packages (from rembg[cli,gpu]) (3.0.0)
Requirement already satisfied: onnxruntime-gpu in /usr/local/lib/python3.10/dist-packages (from rembg[cli,gpu]) (1.16.3)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->rembg[cli,gpu]) (23.2.0)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp->rembg[cli,gpu]) (6.0.4)
✓ Connected to Python 3 Google Compute Engine backend
```

2. Upload file foto yang akan di hapus back ground nya terlebih dahulu, lalu pindahkan file foto yang sudah di upload tadi ke file sample data, seperti berikut :



3. Masukkan kode untuk memanggil install library tadi :

```
from rembg import remove
from PIL import Image
import cv2
```

4. Masukkan kode untuk mengubah file foto dari jpg ke png seperti berikut :

```
[16] input_path = '/content/sample_data/apel.jpg'
      output_path = '/content/sample_data/apel.png'
```

5. Setelah itu, masukkan kode untuk me- remove background file foto yang sudah dimasukkan tadi

```
✓ 5s ▶ with open(input_path, 'rb') as i:
        with open(output_path, 'wb') as o:
            input = i.read()
            output = remove(input)
            o.write(output)
```

6. Masukkan kode untuk mengubah gambar asli menjadi citra deteksi tepi

```
▶ import cv2
import numpy as np
import matplotlib.pyplot as plt

image = cv2.imread("apel.jpg")
image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

canny_output = cv2.Canny(image, 100, 180)

plt.subplot(121), plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.title("gambar asli"), plt.xticks([]), plt.yticks([])
plt.subplot(122), plt.imshow(canny_output, cmap="gray")
plt.title("citra deteksi tepi"), plt.xticks([]), plt.yticks([])
plt.show()
```

PENJELASAN :

- Perintah 'pip install rembg' digunakan untuk menginstal pustaka python bernama "rembg" (remove background). Ini dirancang untuk membantu pengguna dalam menghapus latar belakang (background) dari gambar. Dengan

menggunakan 'rembg', anda dapat secara otomatis menghapus latar belakang pada gambar, meninggalkan hanya objek tanpa latar belakang.

- 'from rembg import remove' : mengimpor fungsi remove dari pustaka 'rembg' yang digunakan untuk menghapus latar belakang gambar
- 'from PIL import image' : mengimpor kelas 'image' dari pustaka 'PIL' (atau 'Pillow'). Umumnya digunakan untuk memanipulasi gambar setelah latar belakangnya dihapus
- 'import cv2' : mengimpor pustaka openCV, yang sering digunakan untuk manipulasi dan pemrosesan gambar, serta banyak tugas pemrosesan citra lainnya.
- 'input_path' adalah path (alamat file) dari gambar yang akan di proses
'output_path' (alamat file) adalah tempat hasil pemrosesan akan disimpan.
- Seperti contoh kode diatas, "input_apel.jpg" adalah gambar yang ingin di proses (dengan latar belakang yang akan dihapus), dan "output_apel.png" adalah tempat hasilnya akan disimpan setelah latar belakang dihapus.
- 'with open (input_path, 'rb') as i:' membuka file yang berada di alamat 'input_path dalam mode baca biner ('rb') kemudian menggunakannya dalam blok 'with'. Variabel yang diberi nama 'i' adalah objek file, Begitu juga dengan yang output.
- 'input = i.read()' membaca seluruh konten file yang akan diakses menggunakan file handle 'i' dan menyimpannya ke dalam variabel 'input'. Begitu juga dengan yang output.
- 'cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)' untuk mengubah gambar berwarna (mode BGR) menjadi citra keabuan (grayscale). Hasilnya disimpan ke variable 'image'.
- 'cv2.Canny(image, 80, 180)' menerapkan deteksi tepi canny pada citra keabuan. Angka 80 dan 180 adalah batas ambang bawah dan atas untuk deteksi tepi.

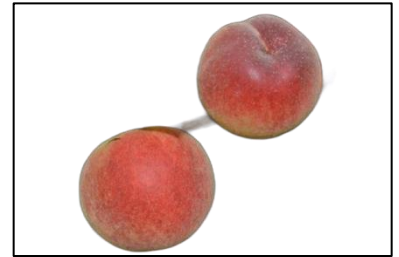
HASIL :

- Citra Wajib :

gambar asli



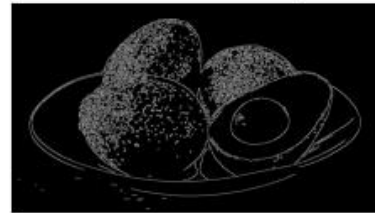
citra deteksi tepi



gambar asli



citra deteksi tepi



gambar asli



citra deteksi tepi

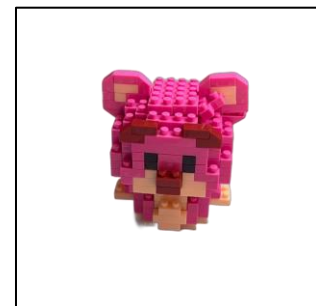
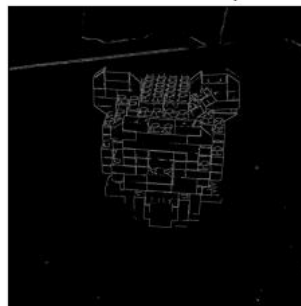


- Citra Bebas :

gambar asli



citra deteksi tepi



gambar asli



citra deteksi tepi

