

**UJIAN AKHIR SEMESTER  
MATA KULIAH  
PRAKTIKUM PENGOLAHAN GAMBAR**

Topik Segmentasi

Gambar atau Foto



**PENYUSUN LAPORAN**



<b>Nama Mahasiswa</b>	<b>NIM</b>	<b>Kelas</b>
M NAUVAL KHATAMI FASHA S	062340833190	1 MIN

**PROGRAM STUDI MANAJEMEN INFORMATIKA  
JURUSAN MANAJEMEN INFORMATIKA  
POLITEKNIK NEGERI SRIWIJAYA  
2024**

Semester : 1 Tanggal : 08 Januari 2024  
Dosen Pengampu : Sulistiyanto, MTI Kelas : 1 MLN  
CP : Mahasiswa/i dapat mengerjakan UAS dengan prosedur yang telah ditentukan.

**Nama Mahasiswa** : M.Nauval Khatami Fasha S  
**NIM** : 062340833190

## **TUJUAN**

1. Memberikan pemahaman singkat dan padat tentang bagaimana program bekerja dalam membaca, menganalisis tepi, dan melakukan segmentasi warna merah pada gambar.
2. Memberikan gambaran yang jelas dan singkat tentang apa yang dicapai oleh kode yang digunakan dan bagaimana cara kerjanya.

## **MATERIAL**

1. Laptop
2. Aplikasi Visual Studio Code, Bahasa Pemrograman Python, PIP Numpy, PIP Matplotlib, dan PIP Open Cv.

## **PROJECT**

1. Setiap capture gambar, diberi penjelasan
  - a). Gambar 1.1 Mendefinisikan fungsi process\_image yang menerima path gambar sebagai argumen.
  - b). Gambar 1.2 Membaca gambar asli dari path yang diberikan.
  - c). Gambar 1.3 :
    - Menggunakan deteksi tepi Canny untuk menemukan tepi pada gambar.
    - Mengonversi gambar ke skala abu-abu sebelum deteksi tepi.
    - Mengonversi gambar tepi ke format RGB.
  - d). Gambar 1.4 :
    - Mengonversi gambar ke ruang warna HSV.
    - Membuat mask dengan menentukan range warna dari gambar yang digunakan.
    - Mengaplikasikan mask pada gambar asli menggunakan operasi bitwise AND.
  - e). Gambar 1.5 :
    - Membuat jendela dengan ukuran 10x5 inch.
    - Menampilkan gambar asli, gambar deteksi tepi, dan gambar hasil segmentasi dalam satu jendela dengan tiga subplot.
    - Menonaktifkan sumbu pada setiap subplot.
    - Menampilkan jendela.

f). Gambar 1.6 Memanggil fungsi process\_image dengan memberikan path gambar 'nama\_gambar.jpg' sebagai argumen.

2. Tulislah dokumentasi kode python untuk segmentasi dan deteksi gambar

a. Process Image

```
def process_image(image_path):
```

Gambar 1.1

b. Membaca Gambar

```
original_image = cv2.imread(image_path)
```

Gambar 1.2

c. Deteksi Tepi

```
edges = cv2.Canny(cv2.cvtColor(original_image, cv2.COLOR_BGR2GRAY), 50, 150)
edge_image = cv2.cvtColor(edges, cv2.COLOR_GRAY2RGB)
```

Gambar 1.3

d. Segmentasi

```
hsv = cv2.cvtColor(original_image, cv2.COLOR_BGR2HSV)
mask = cv2.inRange(hsv, np.array([0, 0, 0]), np.array([10, 400, 255]))
segmented_image = cv2.bitwise_and(original_image, original_image, mask=mask)
```

Gambar 1.4

e. Jendela Output

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

plt.subplot(1, 3, 1)
plt.imshow(cv2.cvtColor(original_image, cv2.COLOR_BGR2RGB))
plt.title('Gambar Asli')
plt.axis('off')

plt.subplot(1, 3, 2)
plt.imshow(edge_image)
plt.title('Deteksi Gambar')
plt.axis('off')

plt.subplot(1, 3, 3)
plt.imshow(cv2.cvtColor(segmented_image, cv2.COLOR_BGR2RGB))
plt.title('Segmentasi')
plt.axis('off')

plt.show()
```

Gambar 1.5

f. Pemanggilan Fungsi Path

```
process_image('apel.jpg')
```

Gambar 1.6

# DOKUMENTASI KODE & HASIL PROGRAM

## A. Citra Wajib “apel.jpg” :

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

# function to perform edge detection on the image
def edge_detection(image):
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    edges = cv2.Canny(gray, 50, 150)
    return cv2.cvtColor(edges, cv2.COLOR_GRAY2RGB)

# function to remove the background from the image
def remove_background(image):
    hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
    lower_red = np.array([0, 0, 0])
    upper_red = np.array([10, 100, 255])
    mask = cv2.inRange(hsv, lower_red, upper_red)
    result = cv2.bitwise_and(image, image, mask = mask)
    return result

# load the original image
original_image = cv2.imread('apel.jpg')

# perform edge detection on the image
edge_image = edge_detection(original_image)

# remove the background from the image
segmented_image = remove_background(original_image)

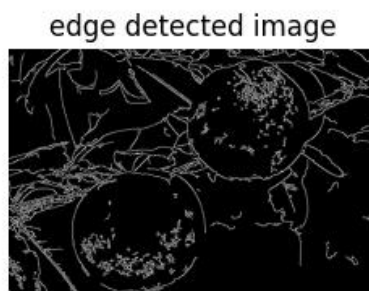
# display the three images in one window
plt.figure(figsize=(10, 5))

# original image
plt.subplot(1, 3, 1)
plt.imshow(cv2.cvtColor(original_image, cv2.COLOR_BGR2RGB))
plt.title("original image")
plt.axis('off')

# image with edges detected
plt.subplot(1, 3, 2)
plt.imshow(edge_image)
plt.title("edge detected image")
plt.axis('off')

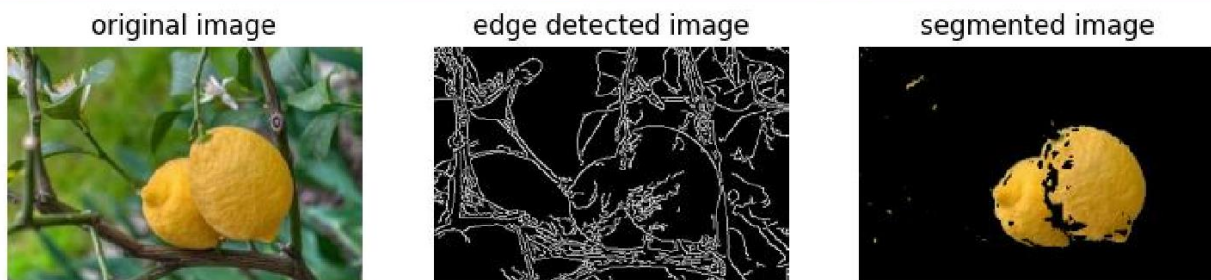
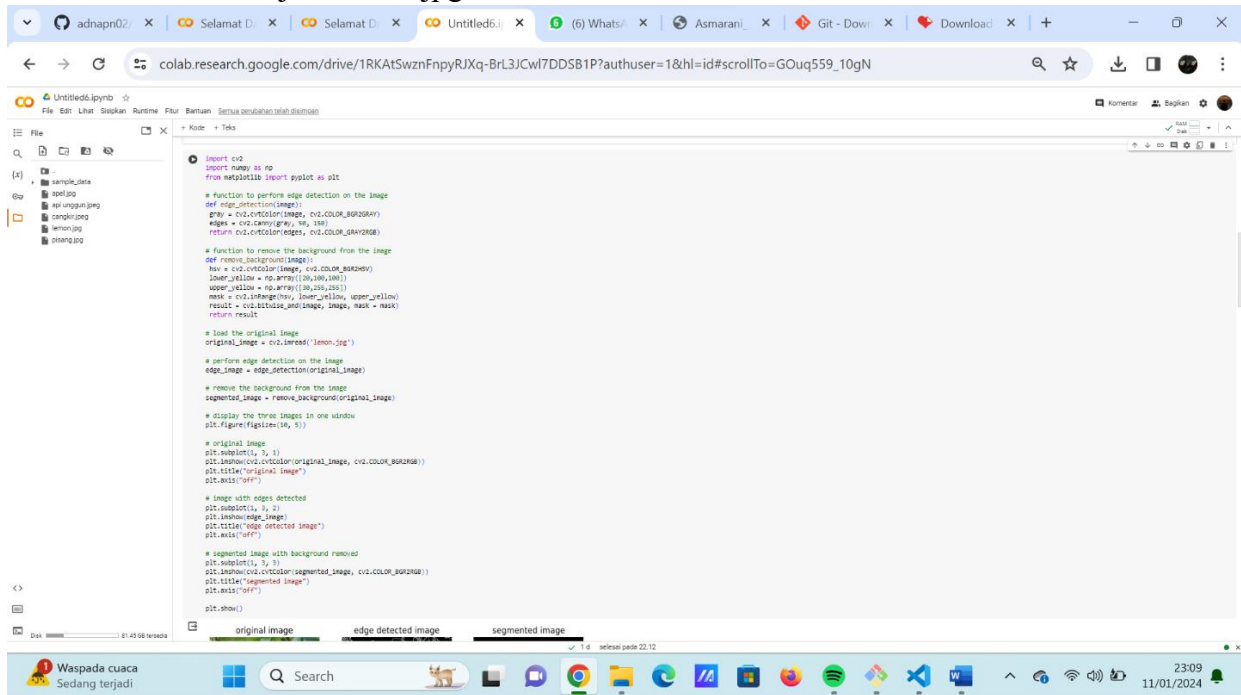
# segmented image with background removed
plt.subplot(1, 3, 3)
plt.imshow(cv2.cvtColor(segmented_image, cv2.COLOR_BGR2RGB))
plt.title("segmented image")
plt.axis('off')

plt.show()
```



Gambar 2.1

## B. Citra Wajib “lemon.jpg” :



Gambar 2.2

### C. Citra Wajib “pisang.jpg”:

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

# function to perform edge detection on the image
def edge_detection(image):
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    edges = cv2.Canny(gray, 50, 150)
    return cv2.cvtColor(edges, cv2.COLOR_GRAY2BGR)

# function to remove the background from the image
def remove_background(image):
    hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
    lower_yellow = np.array([10,100,100])
    upper_yellow = np.array([19,190,190])
    mask = cv2.inRange(hsv, lower_yellow, upper_yellow)
    result = cv2.bitwise_and(image, image, mask = mask)
    return result

# load the original image
original_image = cv2.imread('pisang.jpg')

# perform edge detection on the image
edge_image = edge_detection(original_image)

# remove the background from the image
segmented_image = remove_background(original_image)

# display the three images in one window
plt.figure(figsize=(10, 8))

# original image
plt.subplot(1, 3, 1)
plt.imshow(cv2.cvtColor(original_image, cv2.COLOR_BGR2RGB))
plt.title("original image")
plt.axis("off")

# image with edges detected
plt.subplot(1, 3, 2)
plt.imshow(edge_image)
plt.title("edge detected image")
plt.axis("off")

# segmented image with background removed
plt.subplot(1, 3, 3)
plt.imshow(cv2.cvtColor(segmented_image, cv2.COLOR_BGR2RGB))
plt.title("segmented image")
plt.axis("off")

plt.show()
```

original image



edge detected image



segmented image



Gambar 2.3

#### D. Citra Bebas “dinosaurus.jpeg”:

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

# function to perform edge detection on the image
def edge_detection(image):
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    edges = cv2.Canny(gray, 50, 150)
    return cv2.cvtColor(edges, cv2.COLOR_GRAY2BGR)

# function to remove the background from the image
def remove_background(image):
    hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
    lower_orange = np.array([10,180,40])
    upper_orange = np.array([20,250,255])
    mask = cv2.inRange(hsv, lower_orange, upper_orange)
    result = cv2.bitwise_and(image, image, mask = mask)
    return result

# load the original image
original_image = cv2.imread('dinosaurus.jpeg')

# perform edge detection on the image
edge_image = edge_detection(original_image)

# remove the background from the image
segmented_image = remove_background(original_image)

# display the three images in one window
plt.figure(figsize=(10, 5))

# original image
plt.subplot(1, 3, 1)
plt.imshow(cv2.cvtColor(original_image, cv2.COLOR_BGR2RGB))
plt.title("original image")
plt.axis("off")

# image with edges detected
plt.subplot(1, 3, 2)
plt.imshow(edge_image)
plt.title("edge detected image")
plt.axis("off")

# segmented image with background removed
plt.subplot(1, 3, 3)
plt.imshow(cv2.cvtColor(segmented_image, cv2.COLOR_BGR2RGB))
plt.title("segmented image")
plt.axis("off")

plt.show()
```

original image



edge detected image



segmented image



Gambar 2.4



## E. Citra Bebas “paperbag.jpeg”:

```
Git - Down x pengolah x Dewa x (2) Whats x Salinan da x Selamat D x python - x screenshot x +
colab.research.google.com/?hl=id&authuser=1#scrollTo=eacZDVv_6nrD
Gmail YouTube Maps

import cv2
import numpy as np
from matplotlib import pyplot as plt

# function to perform edge detection on the image
def edge_detection(image):
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    edges = cv2.Canny(gray, 10, 150)
    return cv2.cvtColor(edges, cv2.COLOR_GRAY2RGB)

# function to remove the background from the image
def remove_background(image):
    hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
    lower_red = np.array([0, 4, 40])
    upper_red = np.array([180, 40, 255])
    mask = cv2.inRange(hsv, lower_red, upper_red)
    result = cv2.bitwise_and(image, image, mask = mask)
    return result

# load the original image
original_image = cv2.imread('paperbag.jpeg')

# perform edge detection on the image
edge_image = edge_detection(original_image)

# remove the background from the image
segmented_image = remove_background(original_image)

# display the three images in one window
plt.figure(figsize=(10, 5))

# original image
plt.subplot(1, 3, 1)
plt.imshow(cv2.cvtColor(original_image, cv2.COLOR_BGR2RGB))
plt.title("original image")
plt.axis("off")

# image with edges detected
plt.subplot(1, 3, 2)
plt.imshow(edge_image)
plt.title("edge detected image")
plt.axis("off")

# segmented image with background removed
plt.subplot(1, 3, 3)
plt.imshow(cv2.cvtColor(segmented_image, cv2.COLOR_BGR2RGB))
plt.title("segmented image")
plt.axis("off")

plt.show()
```

original image



edge detected image



segmented image



Gambar 2.5



## ANALISIS

### 1. Membaca Gambar

Gambar asli dibaca menggunakan cv2.imread dari path yang diberikan.

### 2. Deteksi Tepi

- Menggunakan deteksi tepi Canny (cv2.Canny) untuk menemukan tepi pada gambar.
- Mengonversi gambar ke skala abu-abu (cv2.cvtColor(original\_image, cv2.COLOR\_BGR2GRAY)).
- Mengonversi gambar tepi ke format RGB (cv2.cvtColor(edges, cv2.COLOR\_GRAY2RGB)).

### 3. Segmentasi

- Mengonversi gambar ke ruang warna HSV (cv2.cvtColor(original\_image, cv2.COLOR\_BGR2HSV)).
- Membuat mask untuk segmentasi warna sesuai dengan gambar yang digunakan menggunakan cv2.inRange.
- Mengaplikasikan mask pada gambar asli menggunakan operasi bitwise AND (cv2.bitwise\_and).

### 4. Jendela Output

- Membuat jendela matplotlib dengan tiga subplot.
- Subplot pertama menampilkan gambar asli.
- Subplot kedua menampilkan gambar deteksi tepi.
- Subplot ketiga menampilkan gambar hasil segmentasi.

### 5. Menampilkan Jendela Output

Menampilkan jendela matplotlib dengan tiga gambar dalam satu frame.

## KESIMPULAN

1. Kode Program yang ada didalam laporan ini berfungsi melakukan deteksi tepi dan segmentasi warna sesuai dengan gambar yang digunakan.
2. Deteksi tepi membantu mengidentifikasi garis atau kontur pada gambar.
3. Segmentasi warna memisahkan objek berwarna dari latar belakang.
4. Menampilkan ketiga gambar secara bersamaan memberikan pemahaman visual tentang hasil dari dua proses tersebut.