

**LAPORAN UJIAN AKHIR SEMESTER
MATA KULIAH
PENGOLAHAN CITRA DIGITAL**

Topik

Deteksi Gambar & Segmentasi Gambar



PENYUSUN LAPORAN



| Nama Mahasiswa | NIM | Kelas |
|--------------------------|--------------|--------------|
| Muhammad Noor Hidayat | 062340833235 | 1 MIO |

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

UAS Pengolahan Citra Digital 1MIO : Deteksi Gambar dan Segmentasi Gambar Lewat Python

| | | | |
|----------------|----------------------------------------------------------------|---------|-------------------|
| Pertemuan | : UAS | Tanggal | : 10 Januari 2024 |
| Semester | : 1 | Kelas | : 1MIO |
| Dosen Pengampu | : Sulistiyanto, MTI | Tugas | : UAS |
| CP | : Mahasiswa mampu mengerjakan tugas UAS dengan baik dan benar. | | |

Nama Mahasiswa : Muhammad Noor Hidayat
NIM 062340833235

TUJUAN

1. Membuat program Python untuk menampilkan gambar asli, deteksi tepi, dan segmentasi dengan penghapusan latar belakang gambar yang digunakan.

ALAT DAN BAHAN (HW & SW)

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

TUGAS

1. Setiap capture gambar, diberi penjelasan
 - a). Gambar 1.1 :
 - Fungsi ini melakukan deteksi tepi pada gambar menggunakan metode Canny.
 - Gambar diubah ke skala abu-abu untuk mempermudah deteksi tepi.
 - Metode Canny digunakan untuk menemukan tepi dalam gambar.
 - Hasil deteksi tepi dikonversi kembali ke mode warna RGB sebelum dikembalikan.
 - b). Gambar 1.2 :
 - Fungsi ini melakukan segmentasi gambar dengan menghapus latar belakang berdasarkan warna kuning (pisang).
 - Gambar diubah ke ruang warna HSV untuk lebih baik dalam menangkap warna.
 - Batas warna kuning dalam format HSV ditentukan.
 - Mask dibuat untuk menentukan area yang akan dipertahankan (warna kuning).
 - Operasi bitwise digunakan untuk menghapus latar belakang berdasarkan mask.
 - c). Gambar 1.3 :
 - Membaca gambar asli dari file dengan nama Gambar yang digunakan.
 - d). Gambar 1.4 :
 - Memanggil fungsi deteksi tepi untuk mendapatkan gambar dengan tepi yang terdeteksi.
 - e). Gambar 1.5 :
 - Memanggil fungsi penghapusan background untuk mendapatkan gambar dengan latar belakang yang dihapus berdasarkan warna kuning.

f). Gambar 1.6 :

- Membuat jendela dengan tiga subplot menggunakan Matplotlib.
- Menampilkan gambar asli, gambar dengan deteksi tepi, dan gambar dengan segmentasi dalam satu jendela.
- Subplot pertama menampilkan gambar asli, subplot kedua menampilkan gambar dengan deteksi tepi, dan subplot ketiga menampilkan gambar dengan segmentasi dan latar belakang yang dihapus.

g). Gambar 2.1 :

Gambar Kode dan Hasil Program dari Citra Wajib “apel.jpg”.

h). Gambar 2.2 :

Gambar Kode dan Hasil Program dari Citra Wajib “lemon.jpg”.

i). Gambar 2.3 :

Gambar Kode dan Hasil Program dari Citra Wajib “pisang.jpg”.

j). Gambar 2.4 :

Gambar Kode dan Hasil Program dari Citra Tambahan “botol_parfume.jpg”.

k). Gambar 2.5 :

Gambar Kode dan Hasil Program dari Citra Tambahan “foto.jpg”.

2. Tulislah dokumentasi percobaan syntax python dalam mendeteksi tepian dan segmentasi pada gambar.

a. Syntax untuk Medeteksi Gambar

```
4
5 def edge_detection(image):
6     gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
7     edges = cv2.Canny(gray, 10, 10)
8     return cv2.cvtColor(edges, cv2.COLOR_GRAY2RGB)
9
```

Gambar 1.1

b. Syntax untuk Men-Segmentasi Gambar

```
10 def remove_background(image):
11     hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
12     lower_green = np.array([40, 40, 40])
13     upper_green = np.array([80, 255, 255])
14     mask = cv2.inRange(hsv, lower_green, upper_green)
15     result = cv2.bitwise_and(image, image, mask=mask)
16     return result
17
```

Gambar 1.2

c. Membaca Gambar Asli

```
18 original_image = cv2.imread('Centong.jpg')
19
```

Gambar 1.3

- d. Mendeteksi Gambar Asli

```
19
20 edge_image = edge_detection(original_image)
21
```

Gambar 1.4

- e. Men-Segmentasi Gambar Asli

```
21
22 segmented_image = remove_background(original_image)
23
```

Gambar 1.5

- f. Syntax untuk Menampilkan 3 Gambar dalam Satu Jendela Output

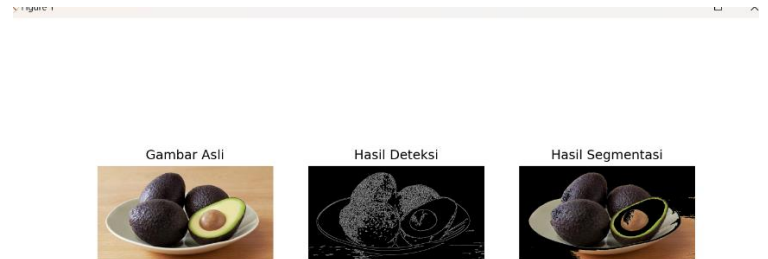
```
23
24 plt.figure(figsize=(10, 5))
25
26 plt.subplot(1, 3, 1)
27 plt.imshow(cv2.cvtColor(original_image, cv2.COLOR_BGR2RGB))
28 plt.title('Gambar Asli')
29 plt.axis('off')
30
31 plt.subplot(1, 3, 2)
32 plt.imshow(edge_image)
33 plt.title('Hasil Deteksi')
34 plt.axis('off')
35
36 plt.subplot(1, 3, 3)
37 plt.imshow(cv2.cvtColor(segmented_image, cv2.COLOR_BGR2RGB))
38 plt.title('Hasil Segmentasi')
39 plt.axis('off')
40
41 plt.show()
42
```

Gambar 1.6

KODE & HASIL PROGRAM:

A. Kode dan Hasil Program dari Citra Wajib “alpukat.jpg” :

```
alpukat.jpg 7...
1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 def edge_detection(image):
6     gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
7     edges = cv2.Canny(gray, 50, 150)
8     return cv2.cvtColor(edges, cv2.COLOR_GRAY2RGB)
9
10 def remove_background(image):
11     hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
12     lower_dark_brown = np.array([0, 0, 0])
13     upper_dark_brown = np.array([300, 355, 195])
14     mask = cv2.inRange(hsv, lower_dark_brown, upper_dark_brown)
15     result = cv2.bitwise_and(image, image, mask=mask)
16     return result
17
18 original_image = cv2.imread('alpukat.jpg')
19
20 edge_image = edge_detection(original_image)
21
22 segmented_image = remove_background(original_image)
23
24 plt.figure(figsize=(10, 5))
25
26 plt.subplot(1, 3, 1)
27 plt.imshow(cv2.cvtColor(original_image, cv2.COLOR_BGR2RGB))
28 plt.title('Gambar Asli')
29 plt.axis('off')
30
31 plt.subplot(1, 3, 2)
32 plt.imshow(edge_image)
33 plt.title('Hasil Deteksi')
34 plt.axis('off')
35
36 plt.subplot(1, 3, 3)
37 plt.imshow(cv2.cvtColor(segmented_image, cv2.COLOR_BGR2RGB))
38 plt.title('Hasil Segmentasi')
39 plt.axis('off')
40
41 plt.show()
```



Gambar 2.1

B. Kode dan Hasil Program dari Citra Wajib “lemon.jpg” :

```
1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 def edge_detection(image):
6     gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
7     edges = cv2.Canny(gray, 50, 150)
8     return cv2.cvtColor(edges, cv2.COLOR_GRAY2RGB)
9
10 def remove_background(image):
11     hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
12     lower_yellow = np.array([20, 100, 100])
13     upper_yellow = np.array([30, 255, 255])
14     mask = cv2.inRange(hsv, lower_yellow, upper_yellow)
15     result = cv2.bitwise_and(image, image, mask=mask)
16     return result
17
18 original_image = cv2.imread('lemon.jpg')
19
20 edge_image = edge_detection(original_image)
21
22 segmented_image = remove_background(original_image)
23
24 plt.figure(figsize=(10, 5))
25
26 plt.subplot(1, 3, 1)
27 plt.imshow(cv2.cvtColor(original_image, cv2.COLOR_BGR2RGB))
28 plt.title('Gambar Asli')
29 plt.axis('off')
30
31 plt.subplot(1, 3, 2)
32 plt.imshow(edge_image)
33 plt.title('Hasil Deteksi')
34 plt.axis('off')
35
36 plt.subplot(1, 3, 3)
37 plt.imshow(cv2.cvtColor(segmented_image, cv2.COLOR_BGR2RGB))
38 plt.title('Hasil Segmentasi')
39 plt.axis('off')
40
41 # Tampilkan jendela
42 plt.show()
```

Gambar Asli



Hasil Deteksi



Hasil Segmentasi



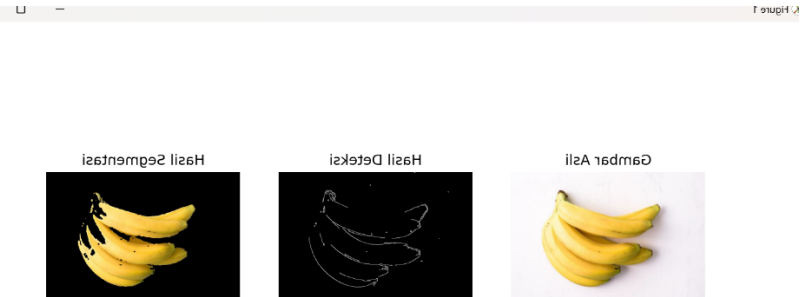
Gambar 2.2

C. Kode dan Hasil Program dari Citra Wajib “pisang.jpg”:

```

1  import cv2
2  import numpy as np
3  import matplotlib.pyplot as plt
4
5  def edge_detection(image):
6      gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
7      edges = cv2.Canny(gray, 50, 150)
8      return cv2.cvtColor(edges, cv2.COLOR_GRAY2RGB)
9
10 def remove_background(image):
11     hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
12     lower_yellow = np.array([20, 100, 100])
13     upper_yellow = np.array([30, 255, 255])
14     mask = cv2.inRange(hsv, lower_yellow, upper_yellow)
15     result = cv2.bitwise_and(image, image, mask=mask)
16     return result
17
18 original_image = cv2.imread('pisang.jpg')
19
20 edge_image = edge_detection(original_image)
21
22 segmented_image = remove_background(original_image)
23
24 plt.figure(figsize=(10, 5))
25
26 plt.subplot(1, 3, 1)
27 plt.imshow(cv2.cvtColor(original_image, cv2.COLOR_BGR2RGB))
28 plt.title('Gambar Asli')
29 plt.axis('off')
30
31 plt.subplot(1, 3, 2)
32 plt.imshow(edge_image)
33 plt.title('Hasil Deteksi')
34 plt.axis('off')
35
36 plt.subplot(1, 3, 3)
37 plt.imshow(cv2.cvtColor(segmented_image, cv2.COLOR_BGR2RGB))
38 plt.title('Hasil Segmentasi')
39 plt.axis('off')
40
41 plt.show()

```



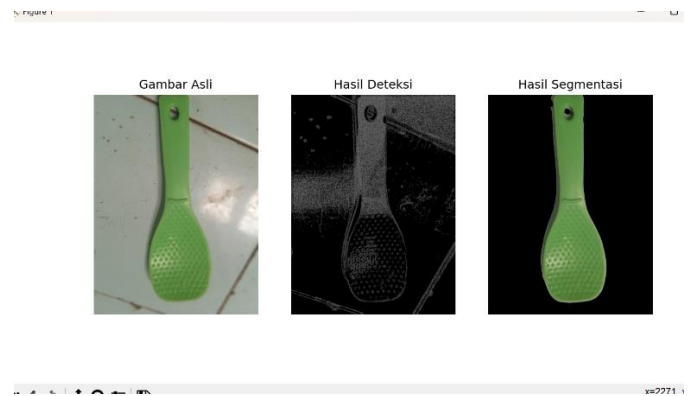
Gambar 2.3

D. Kode dan Hasil Program dari Citra Tambahan “centong.jpg”:

```

1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 def edge_detection(image):
6     gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
7     edges = cv2.Canny(gray, 10, 10)
8     return cv2.cvtColor(edges, cv2.COLOR_GRAY2RGB)
9
10 def remove_background(image):
11     hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
12     lower_green = np.array([40, 40, 40])
13     upper_green = np.array([80, 255, 255])
14     mask = cv2.inRange(hsv, lower_green, upper_green)
15     result = cv2.bitwise_and(image, image, mask=mask)
16     return result
17
18 original_image = cv2.imread('Centong.jpg')
19
20 edge_image = edge_detection(original_image)
21
22 segmented_image = remove_background(original_image)
23
24 plt.figure(figsize=(10, 5))
25
26 plt.subplot(1, 3, 1)
27 plt.imshow(cv2.cvtColor(original_image, cv2.COLOR_BGR2RGB))
28 plt.title('Gambar Asli')
29 plt.axis('off')
30
31 plt.subplot(1, 3, 2)
32 plt.imshow(edge_image)
33 plt.title('Hasil Deteksi')
34 plt.axis('off')
35
36 plt.subplot(1, 3, 3)
37 plt.imshow(cv2.cvtColor(segmented_image, cv2.COLOR_BGR2RGB))
38 plt.title('Hasil Segmentasi')
39 plt.axis('off')
40
41 plt.show()

```



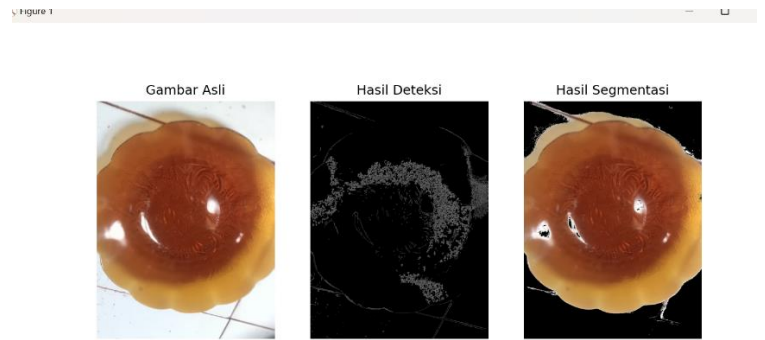
Gambar 2.4

E. Kode dan Hasil Program dari Citra Tambahan “piring.jpg”:

```

Piring.py > edge_detection
1  import cv2
2  import numpy as np
3  import matplotlib.pyplot as plt
4
5  def edge_detection(image):
6      gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
7      edges = cv2.Canny(gray, 10, 50)
8      return cv2.cvtColor(edges, cv2.COLOR_GRAY2RGB)
9
10 def remove_background(image):
11     hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
12     lower_brown = np.array([10, 0, 0])
13     upper_brown = np.array([20, 285, 385])
14     mask = cv2.inRange(hsv, lower_brown, upper_brown)
15     result = cv2.bitwise_and(image, image, mask=mask)
16     return result
17
18 original_image = cv2.imread('Piring.jpg')
19
20 edge_image = edge_detection(original_image)
21
22 segmented_image = remove_background(original_image)
23
24 plt.figure(figsize=(10, 5))
25
26 plt.subplot(1, 3, 1)
27 plt.imshow(cv2.cvtColor(original_image, cv2.COLOR_BGR2RGB))
28 plt.title('Gambar Asli')
29 plt.axis('off')
30
31 plt.subplot(1, 3, 2)
32 plt.imshow(edge_image)
33 plt.title('Hasil Deteksi')
34 plt.axis('off')
35
36 plt.subplot(1, 3, 3)
37 plt.imshow(cv2.cvtColor(segmented_image, cv2.COLOR_BGR2RGB))
38 plt.title('Hasil Segmentasi')
39 plt.axis('off')
40
41 plt.show()
42

```



(Note : foto.jpg dipotret lewat kamera hp, bukan file foto dari kamera profesional.)

Gambar 2.5

Analisis Cara Kerja Fungsi Program/Algoritma:

Deteksi Tepi (Fungsi ``edge_detection``):

- Fungsi menerima input gambar (``image``).
- Gambar diubah ke skala abu-abu menggunakan ``cv2.cvtColor``.
- Deteksi tepi dilakukan menggunakan metode Canny dengan nilai ambang batas 50 dan 150.
- Hasil deteksi tepi dalam skala abu-abu dikonversi kembali ke mode warna RGB menggunakan ``cv2.cvtColor``.
- Gambar dengan tepi yang terdeteksi dikembalikan sebagai output.

Segmentasi Penghapusan Background (Fungsi ``remove_background``):

- Fungsi menerima input gambar (``image``).
- Gambar diubah ke ruang warna HSV menggunakan ``cv2.cvtColor``.
- Nilai batas bawah dan atas untuk warna dari latar belakang yang di hapus dalam format HSV ditentukan.
- Mask dibuat menggunakan ``cv2.inRange`` untuk menentukan area yang akan dipertahankan.
- Operasi bitwise (``cv2.bitwise_and``) digunakan untuk menghapus latar belakang berdasarkan mask.
- Gambar hasil segmentasi dikembalikan sebagai output.

Baca Gambar Asli:

- Gambar asli dibaca menggunakan ``cv2.imread`` dengan nama file(harus disesuaikan dengan nama file/gambar yang yang digunakan).

Deteksi Tepi pada Gambar Asli:

- Fungsi deteksi tepi (``edge_detection``) dipanggil untuk mendapatkan gambar dengan tepi yang terdeteksi.

Penghapusan Background pada Gambar Asli:

- Fungsi penghapusan background (``remove_background``) dipanggil untuk mendapatkan gambar dengan latar belakang yang dihapus.

Tampilkan Ketiga Gambar dalam Satu Jendela:

- Jendela dengan tiga subplot dibuat menggunakan ``plt.figure``.
- Subplot pertama menampilkan gambar asli, subplot kedua menampilkan gambar dengan deteksi tepi, dan subplot ketiga menampilkan gambar dengan segmentasi dan latar belakang yang dihapus.
- Hasilnya ditampilkan menggunakan ``plt.show``.

Hasil dari Kode Program:

- Subplot pertama menampilkan gambar asli.
- Subplot kedua menampilkan gambar dengan deteksi tepi menggunakan metode Canny.
- Subplot ketiga menampilkan gambar dengan segmentasi, di mana latar belakang yang telah dihapus.

KESIMPULAN

Program ini menggunakan deteksi tepi dan segmentasi berbasis warna untuk menghasilkan tiga gambar dalam satu jendela: gambar asli, gambar dengan deteksi tepi, dan gambar segmentasi dengan latar belakang yang dihapus. Fungsi-fungsi tersebut memanfaatkan pustaka OpenCV untuk pemrosesan gambar dan Matplotlib untuk tampilan visual. Program ini dapat berguna dalam pemrosesan gambar untuk menyoroti objek tertentu atau menghapus latar belakang berdasarkan warna.