

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

Topik

Deteksi Gambar & Segmentasi Gambar



PENYUSUN LAPORAN

Nama Mahasiswa	NIM	Kelas
Aji Rausyan F.	062340833229	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 : Aji Rausyan F.
NIM : 062340833229

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 mobil.jpg”.

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

a. Syntax untuk Medeteksi Gambar

```
# Fungsi untuk melakukan deteksi tepi pada gambar
def edge_detection(image):
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    edges = cv2.Canny(gray, 50, 150)
    return cv2.cvtColor(edges, cv2.COLOR_GRAY2RGB)
```

Gambar 1.1

b. Syntax untuk Men-Segmentasi Gambar

```
# Fungsi untuk melakukan segmentasi (hapus background) pada gambar
def remove_background(image):
    hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
    lower_red = np.array([0, 0, 0]) # Nilai batas bawah untuk warna merah dalam format HSV
    upper_red = np.array([10, 400, 255]) # Nilai batas atas untuk warna merah dalam format HSV
    mask = cv2.inRange(hsv, lower_red, upper_red)
    result = cv2.bitwise_and(image, image, mask=mask)
    return result
```

Gambar 1.2

c. Membaca Gambar Asli

```
# Baca gambar asli
original_image = cv2.imread('apel.jpg')
```

Gambar 1.3

- d. Mendeteksi Gambar Asli

```
# Deteksi tepi pada gambar
edge_image = edge_detection(original_image)
```

Gambar 1.4

- e. Men-Segmentasi Gambar Asli

```
# Hapus background pada gambar dengan segmentasi warna merah
segmented_image = remove_background(original_image)
```

Gambar 1.5

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

```
# Tampilkan ketiga gambar dalam satu jendela
plt.figure(figsize=(10, 5))

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

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

# Gambar dengan segmentasi warna merah (background dihapus)
plt.subplot(1, 3, 3)
plt.imshow(cv2.cvtColor(segmented_image, cv2.COLOR_BGR2RGB))
plt.title('Segmentasi')
plt.axis('off')

# Tampilkan jendela
plt.show()
```

Gambar 1.6

KODE & HASIL PROGRAM:

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

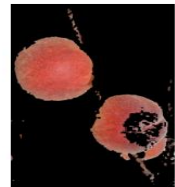
```
Cv > UAS_Pengolahan_Segmentasi > display_image
1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 def display_image(image, title):
6     cv2.imshow(title, image)
7     cv2.waitKey(0)
8     cv2.destroyAllWindows()
9
10 # Fungsi untuk melakukan deteksi tepi pada gambar
11 def edge_detection(image):
12     gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
13     edges = cv2.Canny(gray, 50, 150)
14     return cv2.cvtColor(edges, cv2.COLOR_GRAY2RGB)
15
16 # Fungsi untuk melakukan segmentasi (hapus background) pada gambar
17 def remove_background(image):
18     hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
19     lower_red = np.array([0, 0, 0]) # Nilai batas bawah untuk merah merah dalam format HSV
20     upper_red = np.array([10, 400, 255]) # Nilai batas atas untuk warna merah dalam format HSV
21     mask = cv2.inRange(hsv, lower_red, upper_red)
22     result = cv2.bitwise_and(image, image, mask=mask)
23     return result
24
25 # Baca gambar
26 original_image = cv2.imread('pisang.jpg')
27
28 # Tampilkan gambar asli
29 display_image(original_image, 'Original Image')
30
31 # Deteksi tepi pada gambar
32 edge_image = edge_detection(original_image)
```



Gambar Asli



Deteksi Gambar



Segmentasi

```
Cv > UAS_apel > ...
33
34 # Hapus background pada gambar dengan segmentasi warna merah
35 segmented_image = remove_background(original_image)
36
37 # Tampilkan ketiga gambar dalam satu jendela
38 plt.figure(figsize=(10, 5))
39
40 # Gambar asli
41 plt.subplot(1, 3, 1)
42 plt.imshow(cv2.cvtColor(original_image, cv2.COLOR_BGR2RGB))
43 plt.title('Gambar Asli')
44 plt.axis('off')
45
46 # Gambar dengan deteksi tepi
47 plt.subplot(1, 3, 2)
48 plt.imshow(edge_image)
49 plt.title('Deteksi Gambar')
50 plt.axis('off')
51
52 # Gambar dengan segmentasi warna merah (background dihapus)
53 plt.subplot(1, 3, 3)
54 plt.imshow(cv2.cvtColor(segmented_image, cv2.COLOR_BGR2RGB))
55 plt.title('Segmentasi')
56 plt.axis('off')
57
58 # Tampilkan jendela
59 plt.show()
```

Gambar 2

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

```
Cv > UAS_lemon > edge_detection
1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 def display_image(image, title):
6     cv2.imshow(title, image)
7     cv2.waitKey(0)
8     cv2.destroyAllWindows()
9
10 # Fungsi untuk melakukan deteksi tepi pada gambar
11 def edge_detection(image):
12     gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
13     edges = cv2.Canny(gray, 50, 150)
14     return cv2.cvtColor(edges, cv2.COLOR_GRAY2RGB)
15
16 # Fungsi untuk melakukan segmentasi (hapus background) pada gambar
17 def remove_background(image):
18     hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
19     lower_yellow = np.array([20, 100, 100]) # Nilai batas bawah untuk warna kuning dalam format HSV
20     upper_yellow = np.array([30, 255, 255]) # Nilai batas atas untuk warna kuning dalam format HSV
21     mask = cv2.inRange(hsv, lower_yellow, upper_yellow)
22     result = cv2.bitwise_and(image, image, mask=mask)
23     return result
24
25 # Baca gambar asli
26 original_image = cv2.imread('lemon.jpg')
27
28 # Deteksi tepi pada gambar
29 edge_image = edge_detection(original_image)
30
31 # Hapus background pada gambar dengan segmentasi warna kuning
32 segmented_image = remove_background(original_image)
```

```
Cv > UAS_lemon > display_image
33
34 # Tampilkan ketiga gambar dalam satu jendela
35 plt.figure(figsize=(10, 5))
36
37 # Gambar asli
38 plt.subplot(1, 3, 1)
39 plt.imshow(cv2.cvtColor(original_image, cv2.COLOR_BGR2RGB))
40 plt.title('Gambar Asli')
41 plt.axis('off')
42
43 # Gambar dengan deteksi tepi
44 plt.subplot(1, 3, 2)
45 plt.imshow(edge_image)
46 plt.title('Deteksi Gambar')
47 plt.axis('off')
48
49 # Gambar dengan segmentasi warna kuning (background dihapus)
50 plt.subplot(1, 3, 3)
51 plt.imshow(cv2.cvtColor(segmented_image, cv2.COLOR_BGR2RGB))
52 plt.title('Segmentasi')
53 plt.axis('off')
54
55 # Tampilkan jendela
56 plt.show()
```



Gambar 2.2

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

```
Cv > UAS_pisang > display_image
1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 def display_image(image, title):
6     cv2.imshow(title, image)
7     cv2.waitKey(0)
8     cv2.destroyAllWindows()
9
10 # Fungsi untuk melakukan deteksi tepi pada gambar
11 def edge_detection(image):
12     gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
13     edges = cv2.Canny(gray, 50, 150)
14     return cv2.cvtColor(edges, cv2.COLOR_GRAY2RGB)
15
16 # Fungsi untuk melakukan segmentasi (hapus background) pada gambar
17 def remove_background(image):
18     hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
19     lower_yellow = np.array([20, 100, 100]) # Nilai batas bawah untuk warna kuning dalam format HSV
20     upper_yellow = np.array([30, 255, 255]) # Nilai batas atas untuk warna kuning dalam format HSV
21     mask = cv2.inRange(hsv, lower_yellow, upper_yellow)
22     result = cv2.bitwise_and(image, image, mask=mask)
23     return result
24
25 # Baca gambar asli
26 original_image = cv2.imread('pisang.jpg') # Ganti dengan nama file pisang yang sesuai
27
28 # Deteksi tepi pada gambar
29 edge_image = edge_detection(original_image)
30
31 # Hapus background pada gambar dengan segmentasi warna kuning (pisang)
32 segmented_image = remove_background(original_image)
33
34 # Tampilkan ketiga gambar dalam satu jendela
35 plt.figure(figsize=(10, 5))
36
37 # Gambar asli
38 plt.subplot(1, 3, 1)
39 plt.imshow(cv2.cvtColor(original_image, cv2.COLOR_BGR2RGB))
40 plt.title('Gambar Asli')
41 plt.axis('off')
42
43 # Gambar dengan deteksi tepi
44 plt.subplot(1, 3, 2)
45 plt.imshow(edge_image)
46 plt.title('Deteksi Gambar')
47 plt.axis('off')
48
49 # Gambar dengan segmentasi warna kuning (background dihapus)
50 plt.subplot(1, 3, 3)
51 plt.imshow(cv2.cvtColor(segmented_image, cv2.COLOR_BGR2RGB))
52 plt.title('Segmentasi')
53 plt.axis('off')
54
55 # Tampilkan jendela
56 plt.show()
```



Gambar 2.3

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

```

Cv > UAS_botol > display_image
1  import cv2
2  import numpy as np
3  import matplotlib.pyplot as plt
4
5  def display_image(image, title):
6      cv2.imshow(title, image)
7      cv2.waitKey(0)
8      cv2.destroyAllWindows()
9
10 # Fungsi untuk melakukan deteksi tepi pada gambar
11 def edge_detection(image):
12     gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
13     edges = cv2.Canny(gray, 50, 150)
14     return cv2.cvtColor(edges, cv2.COLOR_GRAY2RGB)
15
16 # Fungsi untuk melakukan segmentasi (hapus background) pada gambar
17 def remove_background(image):
18     hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
19     lower_blue = np.array([90, 50, 50]) # Nilai batas bawah untuk warna biru dalam format HSV
20     upper_blue = np.array([130, 255, 255]) # Nilai batas atas untuk warna biru dalam format HSV
21     mask = cv2.inRange(hsv, lower_blue, upper_blue)
22     result = cv2.bitwise_and(image, image, mask=mask)
23     return result
24
25 # Baca gambar asli
26 original_image = cv2.imread('botol.jpg')
27
28 # Deteksi tepi pada gambar
29 edge_image = edge_detection(original_image)
30
31 # Hapus background pada gambar dengan segmentasi warna biru
32 segmented_image = remove_background(original_image)
33
34 # Tampilkan ketiga gambar dalam satu jendela
35 plt.figure(figsize=(10, 5))
36
37 # Gambar asli
38 plt.subplot(1, 3, 1)
39 plt.imshow(cv2.cvtColor(original_image, cv2.COLOR_BGR2RGB))
40 plt.title('Gambar Asli')
41 plt.axis('off')
42
43 # Gambar dengan deteksi tepi
44 plt.subplot(1, 3, 2)
45 plt.imshow(edge_image)
46 plt.title('Deteksi Gambar')
47 plt.axis('off')
48
49 # Gambar dengan segmentasi warna biru (background dihapus)
50 plt.subplot(1, 3, 3)
51 plt.imshow(cv2.cvtColor(segmented_image, cv2.COLOR_BGR2RGB))
52 plt.title('Segmentasi')
53 plt.axis('off')
54
55 # Tampilkan jendela
56 plt.show()

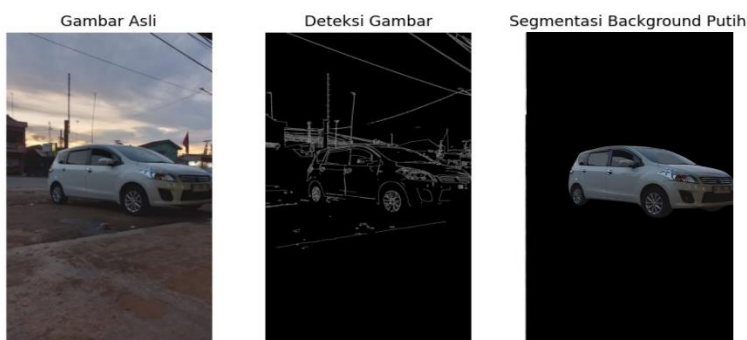
```



Gambar 2.4

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

```
Cv > UAS_mobil > ...
1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 def display_image(image, title):
6     cv2.imshow(title, image)
7     cv2.waitKey(0)
8     cv2.destroyAllWindows()
9
10 # Fungsi untuk melakukan deteksi tepi pada gambar
11 def edge_detection(image):
12     gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
13     edges = cv2.Canny(gray, 50, 150)
14     return cv2.cvtColor(edges, cv2.COLOR_GRAY2RGB)
15
16 # Fungsi untuk melakukan segmentasi (hapus background) pada gambar dengan latar belakang putih
17 def remove_background(image):
18     hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
19     lower_red = np.array([0, 0, 200]) # Nilai batas bawah untuk warna merah dalam format HSV
20     upper_red = np.array([10, 50, 255]) # Nilai batas atas untuk warna merah dalam format HSV
21     mask = cv2.inRange(hsv, lower_red, upper_red)
22
23     # Segmentasi dengan mengganti warna background menjadi putih
24     result = np.ones_like(image) * 255
25     result[mask == 0] = image[mask == 0]
26
27     return result
28
29 # Baca gambar asli
30 original_image = cv2.imread('mobil.jpg')
31
32 # Deteksi tepi pada gambar
33 edge_image = edge_detection(original_image)
34
35 # Hapus background pada gambar dengan segmentasi warna putih
36 segmented_image = remove_background(original_image)
37
38 # Tampilkan ketiga gambar dalam satu jendela
39 plt.figure(figsize=(10, 5))
40
41 # Gambar asli
42 plt.subplot(1, 3, 1)
43 plt.imshow(cv2.cvtColor(original_image, cv2.COLOR_BGR2RGB))
44 plt.title('Gambar Asli')
45 plt.axis('off')
46
47 # Gambar dengan deteksi tepi
48 plt.subplot(1, 3, 2)
49 plt.imshow(edge_image)
50 plt.title('Deteksi Gambar')
51 plt.axis('off')
52
53 # Gambar dengan segmentasi background putih
54 plt.subplot(1, 3, 3)
55 plt.imshow(cv2.cvtColor(segmented_image, cv2.COLOR_BGR2RGB))
56 plt.title('Segmentasi Background Putih')
57 plt.axis('off')
58
59 # Tampilkan jendela
60 plt.show()
61
```



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 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

1. **Tujuan Proyek:**
 - Proyek ini bertujuan untuk melakukan segmentasi citra, khususnya citra buah-buahan, dengan latar belakang yang dihapus menggunakan citra hasil deteksi tepi.
2. **Input Citra:**
 - Terdapat minimal 3 citra buah-buahan yang akan digunakan sebagai citra input untuk proses segmentasi.
 - Selain itu, terdapat 2 citra bebas yang dihasilkan dari jepretan kamera ponsel masing-masing peserta. Citra ini dapat mencakup objek seperti mobil, rumah, hewan, pesawat, dan lainnya.
3. **Proses:**
 - Proses utama dalam proyek ini adalah segmentasi citra. Segmentasi dilakukan dengan menghapus latar belakang citra buah-buahan menggunakan informasi dari citra hasil deteksi tepi.
4. **Alat Pendukung:**
 - Proyek ini memerlukan penggunaan alat atau bahasa pemrograman tertentu untuk melakukan deteksi tepi dan segmentasi citra. Alat-alat seperti OpenCV, TensorFlow, atau PyTorch mungkin digunakan.
5. **Hasil yang Diharapkan:**
 - Hasil akhir proyek ini adalah citra buah-buahan yang telah di-segmentasi, di mana latar belakangnya telah dihapus sehingga hanya objek buah yang tersisa.
6. **Pentingnya Penggunaan Citra Bebas:**
 - Citra bebas yang diambil dari kamera ponsel masing-masing peserta memiliki tujuan untuk menunjukkan keberagaman objek yang dapat diolah menggunakan teknik segmentasi yang sama. Hal ini dapat mencakup objek-objek seperti mobil, rumah, hewan, pesawat, dan lainnya.
7. **Penilaian Kinerja:**
 - Proyek ini kemungkinan akan dinilai berdasarkan akurasi segmentasi, kejelasan hasil, dan keberagaman objek yang dihasilkan dari citra bebas.
8. **Repositori Citra:**
 - Terdapat repo citra-wajib yang berisi citra input yang harus digunakan dalam proyek ini. Repo ini mungkin berisi citra buah-buahan yang akan digunakan sebagai input.
9. **Kesimpulan Umum:**
 - Kesimpulan proyek ini adalah implementasi segmentasi citra pada citra buah-buahan dengan latar belakang dihapus menggunakan citra hasil deteksi tepi, dengan tambahan citra bebas sebagai variasi objek yang dapat diolah.

Pastikan untuk memahami persyaratan proyek dengan baik dan menggunakan alat serta teknik yang sesuai untuk mencapai hasil yang diinginkan.