UJIAN AKHIR SEMESTER

PRAKTIKUM PENGOLAHAN GAMBAR KOMPUTER & FOTOGRAFI

Oleh :

Ersha Kirana (062340833232)

POLITEKNIK NEGERI SRIWIJAYA

JURUSAN MANAJEMEN INFORMATIKA

PALEMBANG

DAFTAR ISI

A. PENJELASAN SOURCE CODE

1. Mengimpor library yang diperlukan, yaitu cv2 (OpenCV) dan NumPy.

```
# Import library yang diperlukan.
import cv2
import numpy as np
```

2. Menggunakan method imread() untuk membaca file citra kedalam variable “citra”.

```
# Membaca file citra, lalu disimpan kedalam variabel "citra".
citra = cv2.imread("apel.jpg")
```

3. Menggunakan method Canny(). Method ini merupakan implementasi dari Metode Canny Edge Detection pada OpenCV. Metode ini terdiri dari 4 langkah. Pertama, Canny akan menghilangkan noise pada citra menggunakan metode *Gaussian Smoothing*. Kedua, Canny akan menghitung gradien pada citra menggunakan *Sobel filter*. Ketiga, Canny akan menerapkan *Non-Max Suppression* untuk menyimpan maxima lokal. Terakhir, Canny akan menerapkan *Hysteresis thresholding* menggunakan nilai dari upper threshold dan lower threshold. Pada program ini, method Canny() digunakan untuk mendeteksi tepi objek pada citra.

```
# Menggunakan metode Canny Edge Detection untuk menghilangkan Noise dengan metode Gaussian Smoothing, dan menerapkan
# threshold dengan 2 threshold values yang dimasukkan sebagai argumen kedalam method cv2.Canny().
tepi_citra = cv2.Canny(citra, 100, 300)
```

4. Mengkonversi citra dari RGB ke HSV.

```
# Mengkonversi citra dari RGB ke HSV.
citraInHSV = cv2.cvtColor(citra, cv2.COLOR_BGR2HSV)
```

5. Menentukan batas maksimal dan batas minimal threshold dengan bantuan library NumPy.

```
# Menentukan batas minimal threshold.
lower_b = np.array([0, 0, 30])

# Menentukan batas maksimal threshold.
upper_b = np.array([200, 60, 300])
```

6. Membuat mask citra menggunakan method inRange().

```
# Membuat mask citra menggunakan method inRange(), dengan variabel "lower_b" sebagai batas minimal threshold dan
# variabel "upper_b" sebagai batas maksimal threshold.
mask_citra = cv2.inRange(citraInHSV, lower_b, upper_b)
```

7. Memisahkan objek dari background menggunakan method bitwise_and()

```
# Memisahkan objek dari background dalam citra menggunakan method bitwise_and() dengan bantuan masking dari "mask_citra"
citra_segmentasi = cv2.bitwise_and(citra, citra, mask=~mask_citra)
```

8. Menampilkan hasil.

```
# Menampilkan Citra asli, Citra deteksi tepi dan Citra hasil segmentasi
cv2.imshow('Gambar asli', citra)
cv2.imshow('Citra deteksi tepi', tepi_citra)
cv2.imshow('Citra hasil segmentasi', citra_segmentasi)

cv2.waitKey(0)
cv2.destroyAllWindows()

```

9. Full Code.

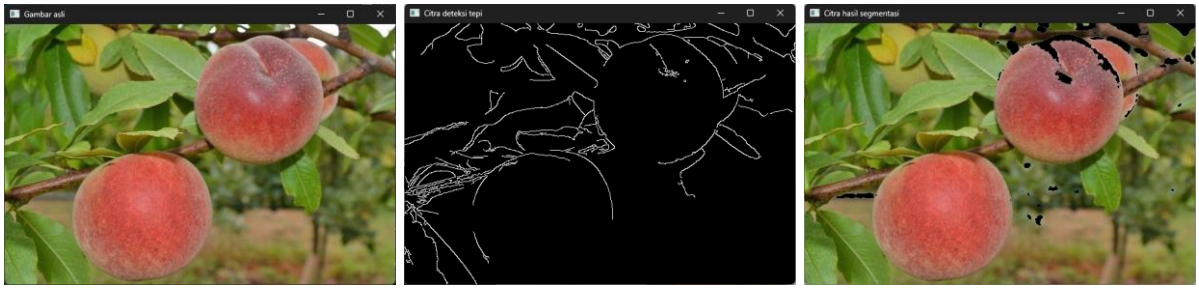
citracv.py

```
# Import library yang diperlukan.
import cv2
import numpy as np

# Membaca file citra, lalu disimpan kedalam variabel "citra".
citra = cv2.imread("apel.jpg")

# Menggunakan metode Canny Edge Detection untuk menghilangkan Noise dengan metode Gaussian Smoothing, dan menerapkan
# threshold dengan 2 threshold values yang dimasukkan sebagai argumen kedalam method cv2.Canny().
tepi_citra = cv2.Canny(citra, 100, 300)

# Mengkonversi citra dari RGB ke HSV.
citraInHSV = cv2.cvtColor(citra, cv2.COLOR_BGR2HSV)

# Menentukan batas minimal threshold.
lower_b = np.array([0, 0, 30])

# Menentukan batas maksimal threshold.
upper_b = np.array([200, 60, 300])

# Membuat mask citra menggunakan method inRange(), dengan variabel "lower_b" sebagai batas minimal threshold dan
# variabel "upper_b" sebagai batas maksimal threshold.
mask_citra = cv2.inRange(citraInHSV, lower_b, upper_b)

# Memisahkan objek dari background dalam citra menggunakan method bitwise_and() dengan bantuan masking dari "mask_citra"
citra_segmentasi = cv2.bitwise_and(citra, citra, mask=~mask_citra)

# Menampilkan Citra asli, Citra deteksi tepi dan Citra hasil segmentasi
cv2.imshow('Gambar asli', citra)
cv2.imshow('Citra deteksi tepi', tepi_citra)
cv2.imshow('Citra hasil segmentasi', citra_segmentasi)

cv2.waitKey(0)
cv2.destroyAllWindows()

```

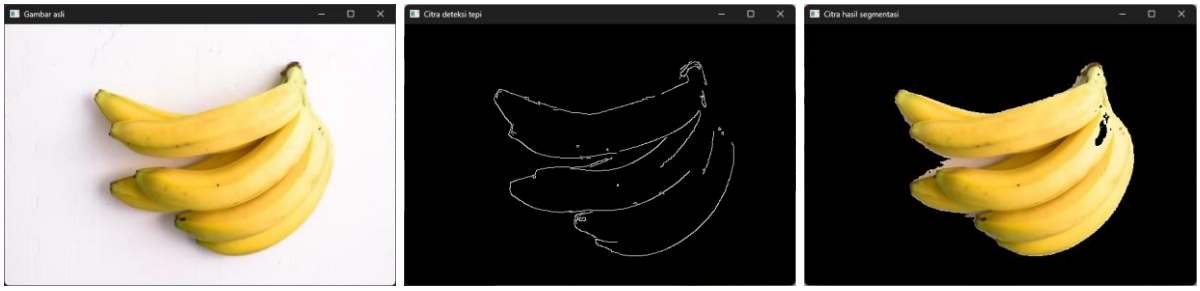
B. HASIL OUTPUT

1. Citra Wajib

    a. Apel



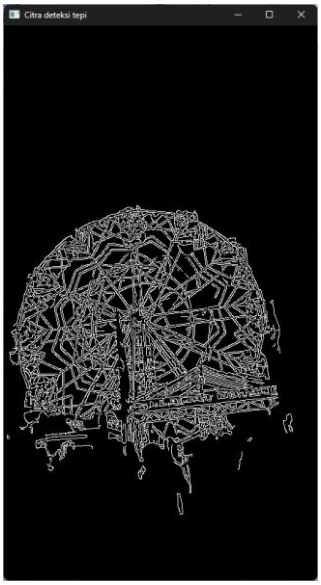
    b. Pisang



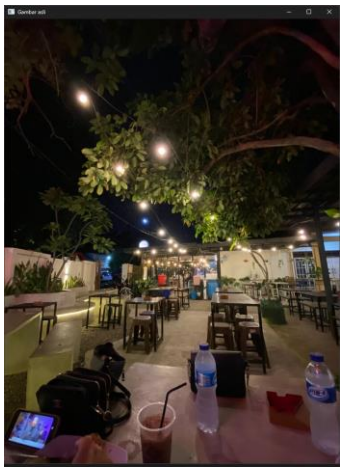
    c. Alpukat

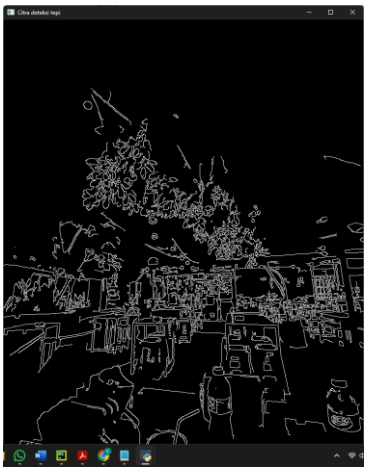
2. Citra Bebas

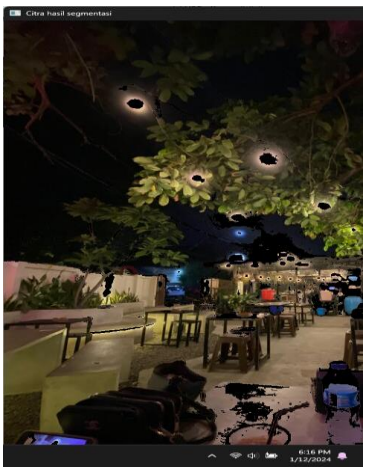
a. Kincir Angin



b. Kafe







C. ALGORITMA PROGRAM

1. Pertama program akan mengimpor seluruh atribut dan method dari library OpenCV dan NumPy.

2. Kemudian, program menggunakan method imread() untuk membaca file citra yang dimasukkan sebagai argumen (“apel.jpg” pada screenshoot), lalu file citra ini akan disimpan pada variabel “citra”.

3. Selanjutnya, program menggunakan method Canny() dengan memasukkan argumen variabel “citra” sebagai input image, 100 sebagai nilai lower threshold dan 300 sebagai nilai upper threshold. Hasilnya disimpan pada variabel “tepi_citra”

4. Lalu program menggunakan method cvtColor() dengan argumen variabel “citra” sebagai input image dan “cv2.COLOR_BG2HSV” sebagai argumen untuk mengubah mode citra dari RGB ke HSV. Lalu citra yang sudah dikonversi disimpan pada variabel “citraInHSV”.

5. Program mengambil nilai array [0, 0, 30], lalu disimpan kedalam variabel “lower_b”.

6. Program mengambil nilai array [200, 60, 300], lalu disimpan kedalam variabel “upper_b”.

7. Menggunakan method inRange() dengan variabel “citraInHSV” sebagai input image, variabel “lower_b” sebagai batas minimal threshold dan variabel “upper_b” sebagai batas maksimal threshold. Proses ini dilakukan untuk membuat masking citra yang disimpan pada variabel “mask_citra”.

8. Program menggunakan method bitwise_and() dengan variabel “citra” sebagai source 1 dan source 2, lalu variabel “mask_citra” sebagai argumen untuk parameter “mask”.

9. Program membuat GUI untuk menampilkan gambar dalam variabel “citra” dengan title “Gambar asli”.

10. Program membuat GUI untuk menampilkan gambar dalam variabel “tepi_citra” dengan title “Citra deteksi tepi”.

11. Program membuat GUI untuk menampilkan gambar dalam variabel “citra_segmentasi” dengan title “Citra hasil segmentasi”.

12. waitKey(0) berfungsi agar program akan menunggu input dari sebuah key pada keyboard sebelum menghilangkan jendela GUI.

13. destroAllWindows() berfungsi untuk menghilangkan semua jendela GUI setelah semua perintah sebelumnya terjalani