Nama : SALDIN

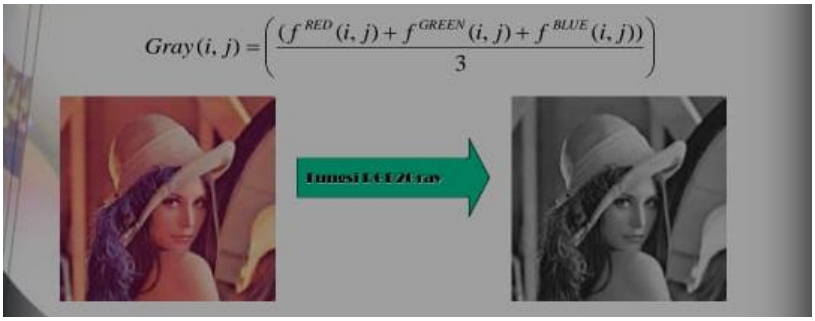
NIM : 22650177

RUMUS DAN ALGORITMA

1. **Transformasi Warna**

Transformasi warna mengubah nilai warna setiap piksel dalam gambar. Dengan menggunakan pustaka **OpenCV** atau **Pillow**, kita bisa melakukan berbagai transformasi warna, seperti konversi ke grayscale, sepia, atau negatif. Berikut adalah beberapa contoh transformasi

a. Grayscale




```python
import cv2

# Memuat gambar
image = cv2.imread('path_to_image.jpg')
# Konversi ke grayscale
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
# Menyimpan hasil
cv2.imwrite('gray_image.jpg', gray_image)
```

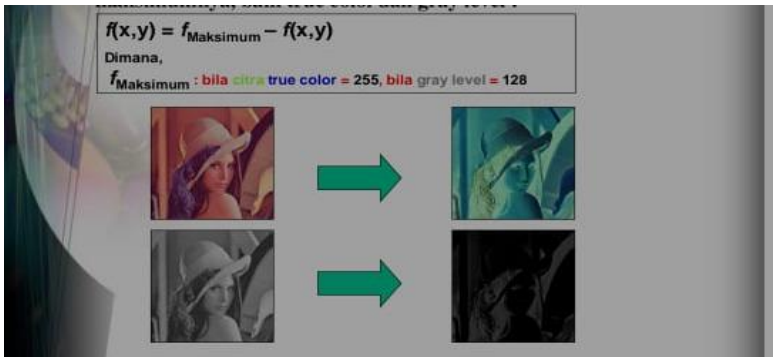
b. Sepia

Efek sepia memberikan nuansa warna cokelat tua seperti foto-foto lama. Ini bisa dilakukan dengan mengalikan matriks warna dengan filter sepia khusus.

```python
import cv2
import numpy as np

# Memuat gambar
image = cv2.imread('path_to_image.jpg')

# Membuat matriks sepia
sepia_filter = np.array([[0.272, 0.534, 0.131],
                         [0.349, 0.686, 0.168],
                         [0.393, 0.769, 0.189]])
sepia_image = cv2.transform(image, sepia_filter)

# Menjaga nilai dalam rentang yang valid
sepia_image = np.clip(sepia_image, 0, 255)

# Menyimpan hasil
cv2.imwrite('sepia_image.jpg', sepia_image)
```

c. Negatif

Transformasi negatif membalikkan warna gambar dengan mengurangi nilai setiap piksel dari 255.



```python
import cv2

# Memuat gambar
image = cv2.imread('path_to_image.jpg')

# Mengubah gambar menjadi negatif
negative_image = 255 - image

# Menyimpan hasil
cv2.imwrite('negative_image.jpg', negative_image)
```

d. Peningkatan Kontras dengan CLAHE (Contrast Limited Adaptive Histogram Equaliztion)

Peningkatan kontras dapat dilakukan untuk memperjelas detail dalam gambar. CLAHE adalah metode untuk memperbaiki kontras dengan cara adaptif.

```python
import cv2

# Memuat gambar dalam grayscale
image = cv2.imread('path_to_image.jpg', 0)

# Membuat objek CLAHE
clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))
contrast_image = clahe.apply(image)

# Menyimpan hasil
cv2.imwrite('contrast_image.jpg', contrast_image)
```

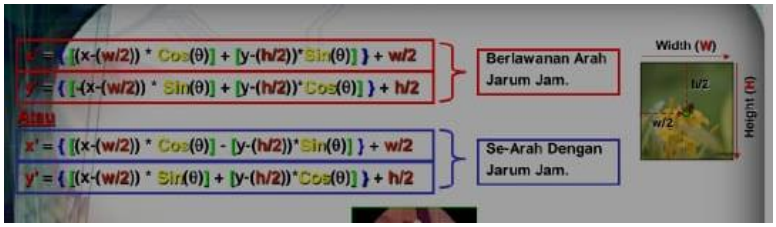
Penjelasan Singkat:

- **Grayscale**: Mengubah gambar menjadi hitam-putih dengan menghilangkan informasi warna.
- **Sepia**: Memberikan nuansa foto klasik dengan warna cokelat.
- **Negatif**: Membalikkan setiap nilai warna piksel.
- **Peningkatan Kontras**: Memperjelas perbedaan antara area terang dan gelap.

Dengan berbagai transformasi warna ini, kita dapat mengubah tampilan gambar sesuai dengan kebutuhan visual atau artistik kita.

2. Rotasi

Rotasi gambar berarti memutar gambar pada titik tertentu, biasanya pada titik tengah gambar. Kita bisa melakukan rotasi dengan sudut tertentu (misalnya 90°, 180°) menggunakan OpenCV. Rumus:



a. Contoh Rotasi 90°

```python
import cv2

# Memuat gambar
image = cv2.imread('path_to_image.jpg')
# Mendapatkan dimensi gambar
(h, w) = image.shape[:2]
# Menentukan titik pusat gambar
center = (w // 2, h // 2)
# Mengatur rotasi sebesar 90 derajat
matrix = cv2.getRotationMatrix2D(center, 90, 1.0)
rotated_image = cv2.warpAffine(image, matrix, (w, h))
# Menyimpan hasil
cv2.imwrite('rotated_image.jpg', rotated_image)
```

b. Contoh Rotasi 45°

```python
import cv2

# Memuat gambar
image = cv2.imread('path_to_image.jpg')
# Mendapatkan dimensi gambar
(h, w) = image.shape[:2]
# Menentukan titik pusat gambar
center = (w // 2, h // 2)
# Mengatur rotasi sebesar 45 derajat
matrix = cv2.getRotationMatrix2D(center, 45, 1.0)
rotated_image_45 = cv2.warpAffine(image, matrix, (w, h))
# Menyimpan hasil
cv2.imwrite('rotated_image_45.jpg', rotated_image_45)
```

c. Contoh Rotasi 180°

Untuk rotasi 180°, kita dapat melakukannya dengan mengubah sudut rotasi pada getRotationMatrix2D.

```python
import cv2

# Memuat gambar
image = cv2.imread('path_to_image.jpg')
# Mendapatkan dimensi gambar
(h, w) = image.shape[:2]
# Menentukan titik pusat gambar
center = (w // 2, h // 2)
# Mengatur rotasi sebesar 180 derajat
matrix = cv2.getRotationMatrix2D(center, 180, 1.0)
rotated_image_180 = cv2.warpAffine(image, matrix, (w, h))
# Menyimpan hasil
cv2.imwrite('rotated_image_180.jpg', rotated_image_180)
```

Penjelasan Lebih Lanjut:

- **cv2.getRotationMatrix2D(center, angle, scale)**:
  - **center**: Titik pusat rotasi.
  - **angle**: Sudut rotasi dalam derajat (misalnya 90, 180, atau sudut lainnya).
  - **scale**: Faktor skala, 1.0 berarti tidak ada perubahan skala.
- **cv2.warpAffine(image, matrix, (w, h))**:
  - **image**: Gambar asli yang akan diputar.
  - **matrix**: Matriks transformasi hasil dari getRotationMatrix2D.
  - **(w, h)**: Ukuran gambar output setelah rotasi.

Dengan memodifikasi nilai **angle** dan **scale**, kita bisa merotasi gambar pada sudut mana pun dan bahkan mengubah ukurannya sesuai keperluan.

3. Scaling

Scaling atau penskalaan bisa dilakukan dengan memperbesar (upscale) atau memperkecil (downscale) gambar. Fungsi cv2.resize pada OpenCV memungkinkan kita menentukan faktor penskalaan untuk kedua dimensi gambar: lebar (fx) dan tinggi (fy). Kita juga bisa menentukan metode interpolasi yang digunakan untuk penskalaan.

a. Scaling dengan Faktor 2x

```python
import cv2

# Memuat gambar
image = cv2.imread('path_to_image.jpg')
# Scaling gambar (misal, 2x lebih besar)
scaled_image = cv2.resize(image, None, fx=2.0, fy=2.0, interpolation=cv2.INTER_LINEAR)
# Menyimpan hasil
cv2.imwrite('scaled_image.jpg', scaled_image)
```

b. Memperkecil Gambar dengan Faktor 0.5

Memperkecil ukuran gambar dapat dilakukan dengan mengatur nilai fx dan fy menjadi kurang dari 1.0.

```python
import cv2

# Memuat gambar
image = cv2.imread('path_to_image.jpg')
# Scaling gambar (misal, 0.5x lebih kecil)
scaled_down_image = cv2.resize(image, None, fx=0.5, fy=0.5, interpolation=cv2.INTER_AREA)
# Menyimpan hasil
cv2.imwrite('scaled_down_image.jpg', scaled_down_image)
```

c. Memperbesar Gambar dengan Ukuran Spesifik

Selain menggunakan faktor penskalaan (fx, fy), kita bisa menentukan ukuran gambar output dengan parameter (width, height).

```python
import cv2

# Memuat gambar
image = cv2.imread('path_to_image.jpg')
# Mengatur ukuran output
new_width = 800
new_height = 600
scaled_to_size_image = cv2.resize(image, (new_width, new_height), interpolation=cv2.INTER_CUBIC)
# Menyimpan hasil
cv2.imwrite('scaled_to_size_image.jpg', scaled_to_size_image)
```

Pilihan Metode Interpolasi

Pada penskalaan, pemilihan metode interpolasi sangat penting untuk menjaga kualitas gambar:

- **cv2.INTER_LINEAR**: Digunakan untuk penskalaan ukuran yang lebih kecil (default).
- **cv2.INTER_CUBIC**: Memberikan hasil yang lebih halus, cocok untuk memperbesar gambar.

- **cv2.INTER_AREA**: Sangat baik untuk mengecilkan ukuran gambar karena menghasilkan hasil yang lebih halus tanpa banyak noise.
- **cv2.INTER_NEAREST**: Proses cepat tetapi kualitas rendah, cocok untuk gambar yang tidak memerlukan detail tinggi.

Dengan memilih interpolasi yang sesuai, kita dapat mengontrol kualitas hasil penskalaan sesuai kebutuhan.

4. Flipping

Flipping adalah transformasi yang mencerminkan gambar, baik secara horizontal, vertikal, atau keduanya. Fungsi cv2.flip digunakan untuk melakukan transformasi ini, dan flipCode menentukan arah pembalikan. Rumus:

Pilihan flipCode:

- **flipCode = 1** : Membalikkan gambar secara horizontal (kiri ke kanan).
- **flipCode = 0** : Membalikkan gambar secara vertikal (atas ke bawah).
- **flipCode = -1** : Membalikkan gambar secara horizontal dan vertikal (efek mirroring keempat kuadran).

a. Membalikkan Gambar Secara Horizontal

```python
import cv2

# Memuat gambar
image = cv2.imread('path_to_image.jpg')
# Membalikkan gambar secara horizontal
flipped_image = cv2.flip(image, 1)
# Menyimpan hasil
cv2.imwrite('flipped_image.jpg', flipped_image)
```

b. Membalikkan Gambar Secara Vertikal

```python
import cv2

# Memuat gambar
image = cv2.imread('path_to_image.jpg')
# Membalikkan gambar secara vertikal
flipped_vertical = cv2.flip(image, 0)
# Menyimpan hasil
cv2.imwrite('flipped_vertical.jpg', flipped_vertical)
```

c. Membalikkan Gambar Secara Horizontal dan Vertikal

```python
import cv2

# Memuat gambar
image = cv2.imread('path_to_image.jpg')
# Membalikkan gambar secara horizontal dan vertikal
flipped_both = cv2.flip(image, -1)
# Menyimpan hasil
cv2.imwrite('flipped_both.jpg', flipped_both)
```

Penjelasan Fungsi cv2.flip

- cv2.flip(image, flipCode): Fungsi ini membalikkan gambar sesuai dengan kode flip yang diberikan.
    - **image**: Gambar input yang ingin dibalikkan.

- **flipCode**: Menentukan arah flipping (1 untuk horizontal, 0 untuk vertikal, -1 untuk keduanya).

Dengan flipping, kita bisa melakukan berbagai efek visual yang bisa berguna dalam pembuatan cermin gambar atau efek simetris dalam pemrosesan gambar.