Nama : RASLAN

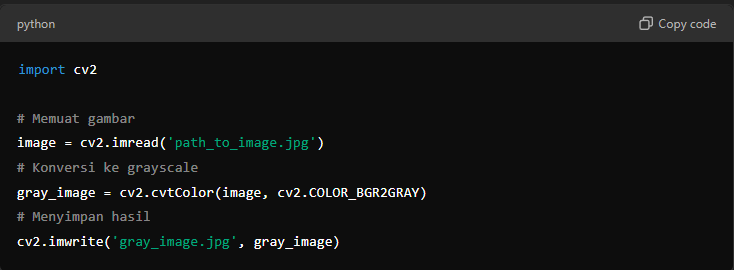
NIM : 22650178

**RUMUS DAN ALGORITMA**

1. **Transformasi Warna**

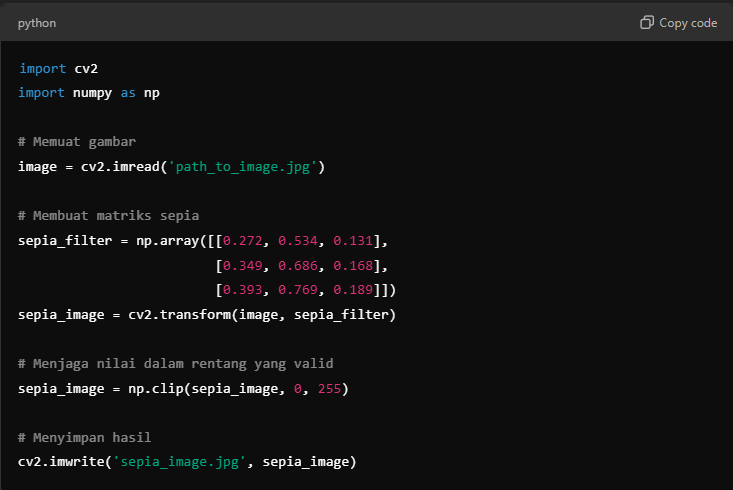
Transformasi warna mengubah nilai warna setiap piksel dalam gambar. Dengan menggunakan pustaka **OpenCV** atau **Pillow**, kita bisa melakukan berbagai transformasi warna, seperti konversi ke grayscale, sepia, atau negatif. Berikut adalah beberapa contoh transformasi

1. Grayscale



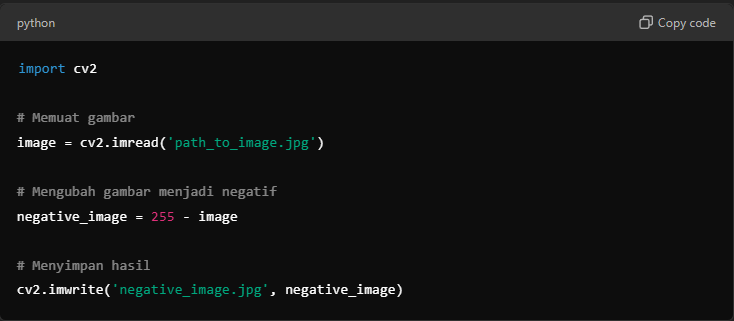
1. Sepia

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



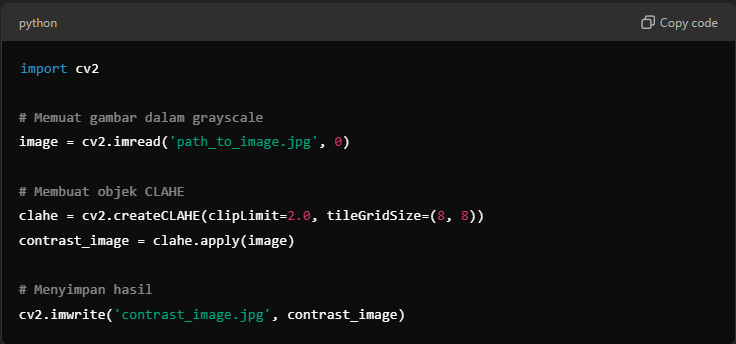
1. Negatif

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



1. 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.



**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.

1. **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.

1. Contoh Rotasi 90°

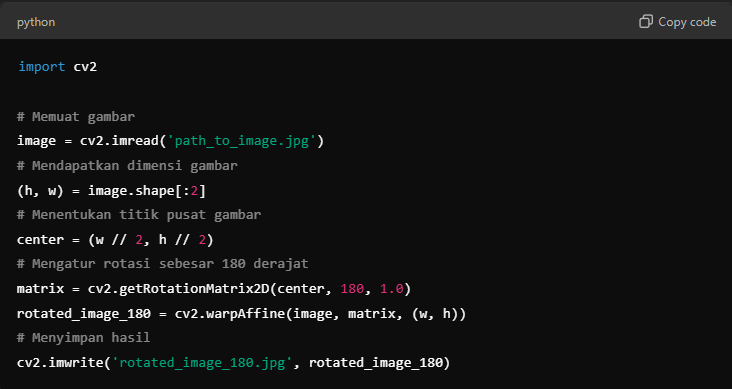


1. Contoh Rotasi 45°



1. Contoh Rotasi 180°

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



**Penjelasan Lebih Lanjut:**

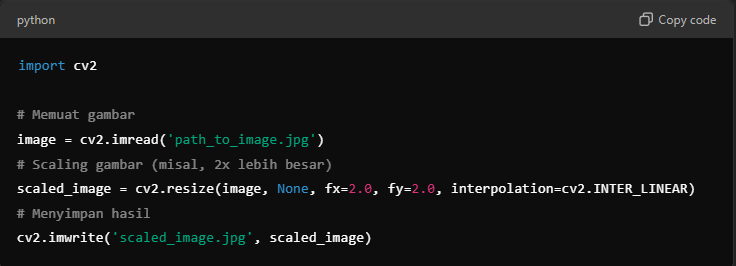
* **cv2.getRotationMatrix2D(center, angle, scale)**:
  + **center**: Titik pusat rotasi.
  + **angle**: Sudut rotasi dalam derajat (misalnya 90, 180, atau sudut lainnya).
  + **scale**: Faktor skala, 1.0 berarti tidak ada perubahan skala.
* **cv2.warpAffine(image, matrix, (w, h))**:
  + **image**: Gambar 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.

1. **Scaling**

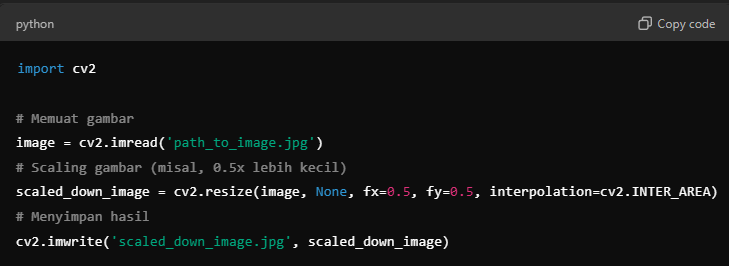
Scaling atau penskalaan bisa dilakukan dengan memperbesar (upscale) atau memperkecil (downscale) gambar. Fungsi cv2.resize pada OpenCV memungkinkan kita menentukan faktor penskalaan untuk kedua dimensi gambar: lebar (fx) dan tinggi (fy). Kita juga bisa menentukan metode interpolasi yang digunakan untuk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

1. Scaling dengan Faktor 2x



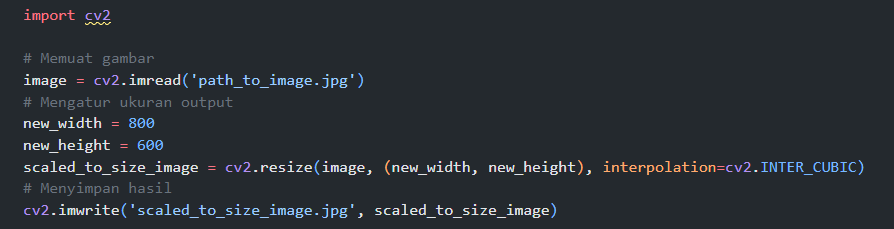
1. Memperkecil Gambar dengan Faktor 0.5

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



1. Memperbesar Gambar dengan Ukuran Spesifik

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



**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.

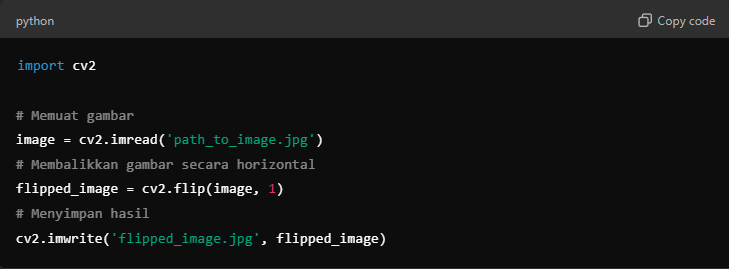
1. **Flipping**

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.

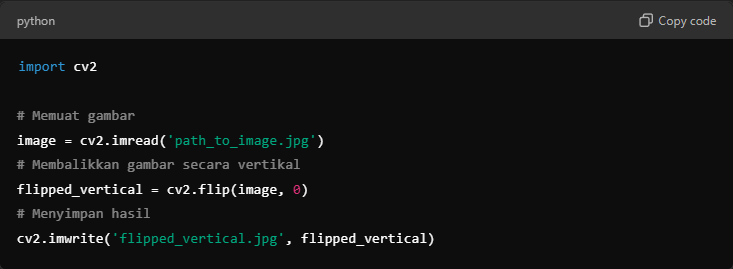
**Pilihan flipCode:**

* **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).

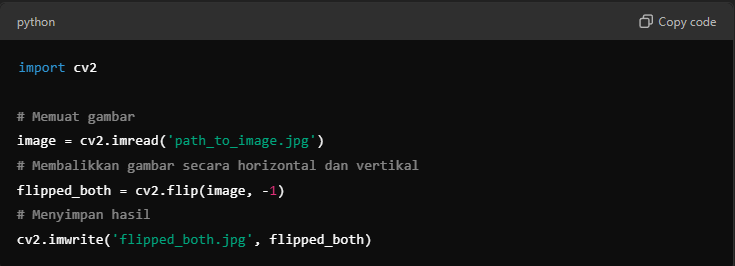
1. Membalikkan Gambar Secara Horizontal



1. Membalikkan Gambar Secara Vertikal



1. Membalikkan Gambar Secara Horizontal dan Vertikal



**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.