import cv2: Ini adalah perintah untuk mengimpor pustaka OpenCV (Open Source Computer Vision Library) ke dalam program Anda. OpenCV adalah pustaka populer yang digunakan untuk berbagai tugas pengolahan citra dan penglihatan komputer.

import numpy as np: Ini mengimpor pustaka NumPy ke program Anda dengan memberinya alias np. NumPy adalah pustaka Python yang kuat untuk komputasi numerik, yang sering digunakan dalam pengolahan citra untuk melakukan operasi matriks dan array.

from matplotlib import pyplot as plt: Ini mengimpor modul pyplot dari pustaka Matplotlib dengan memberinya alias plt. Matplotlib adalah pustaka visualisasi data Python yang sering digunakan untuk membuat plot grafik. Dengan menggunakan pyplot, Anda dapat membuat grafik dan plot citra dengan mudah.

A screenshot of a computer

Description automatically generatedDengan mengimpor ketiga pustaka ini, Anda dapat menggunakan berbagai fungsi dan fitur yang disediakan oleh masing-masing untuk memproses, menganalisis, dan memvisualisasikan citra.

Perintah %matplotlib inline adalah perintah khusus yang digunakan dalam lingkungan Jupyter Notebook atau IPython untuk menampilkan plot Matplotlib secara langsung di dalam notebook, tepat di bawah sel kode yang menghasilkannya. Ini memungkinkan Anda untuk melihat plot secara langsung tanpa perlu memanggil fungsi plt.show().

A screenshot of a computer

Description automatically generatedKetika Anda menjalankan kode di lingkungan Jupyter Notebook atau IPython dengan %matplotlib inline di atas kode yang menghasilkan plot, Matplotlib akan mengatur backend-nya untuk menampilkan plot secara langsung di notebook. Ini sangat berguna saat Anda melakukan eksplorasi data atau prototyping, karena memungkinkan Anda melihat hasil visualisasi secara langsung setelah menjalankan kode.

Program ini membaca sebuah gambar dengan nama file 'nabilalkaf.jpeg' menggunakan fungsi imread dari modul matplotlib.image dan menyimpannya dalam variabel color\_image. Kemudian, gambar tersebut ditampilkan menggunakan fungsi imshow dari modul matplotlib.pyplot, yang menampilkan gambar sebagai plot dalam notebook atau dalam jendela pop-up tergantung pada lingkungan tempat Anda menjalankan kode ini. Dengan demikian, Anda dapat melihat gambar 'nabilalkaf.jpeg' yang dimuat dan ditampilkan di lingkungan tempat Anda menjalankan kode tersebut.

A screenshot of a computer

Description automatically generated

Program ini membagi gambar warna menjadi tiga saluran warna: merah (R), hijau (G), dan biru (B) menggunakan indeksing pada masing-masing saluran warna dalam gambar. Setelah itu, program membuat sebuah subplot yang terdiri dari empat gambar:

1. \*Gambar Asli\*: Menampilkan gambar asli dalam subplot pertama.

2. \*Merah (R)\*: Menampilkan saluran warna merah dalam subplot kedua.

3. \*Hijau (G)\*: Menampilkan saluran warna hijau dalam subplot ketiga.

4. \*Biru (B)\*: Menampilkan saluran warna biru dalam subplot keempat.

Setiap subplot diberi judul sesuai dengan saluran warna yang ditampilkan di dalamnya dan disetel agar tidak menampilkan sumbu (axis) dengan menggunakan axis('off'). Ukuran subplot ditentukan dengan figsize=(20,10) agar plot terlihat dengan jelas. Subplot-subplot ini ditampilkan dalam satu bingkai gambar menggunakan plt.subplots(1, 4, ...) yang menghasilkan empat subplot dalam satu baris.

Dua baris pertama kode yang Anda berikan mengambil gambar dengan nama file 'nabilalkaf.jpeg' menggunakan OpenCV (cv2.imread) dan kemudian mengubah skema warnanya menjadi RGB menggunakan cv2.cvtColor(color\_image, cv2.COLOR\_BGR2RGB). Dengan demikian, gambar tersebut sekarang siap ditampilkan dengan benar menggunakan Matplotlib. Jika Anda ingin melanjutkan dengan kode sebelumnya, Anda dapat menggunakan A screenshot of a computer

Description automatically generatedvariabel color\_image yang baru diubah warnanya.

A close up of a logo

Description automatically generated

Kode tersebut menghitung histogram untuk masing-masing saluran warna (merah, hijau, biru) serta untuk seluruh gambar warna. Berikut adalah penjelasan untuk setiap baris kode:

1. histogram\_biru: Menghitung histogram untuk saluran warna biru (b) dengan menggunakan cv2.calcHist(). Argumen pertama adalah daftar saluran warna yang ingin dihitung histogramnya, di sini hanya saluran warna biru. Argumen kedua adalah indeks saluran warna (dalam hal ini 0 untuk saluran biru), argumen ketiga adalah maska (None dalam hal ini artinya tidak ada maska), argumen keempat adalah jumlah bin histogram (256 dalam hal ini), dan argumen kelima adalah rentang nilai piksel (0 sampai 255 untuk nilai warna dalam skala RGB).

2. histogram\_hijau: Menghitung histogram untuk saluran warna hijau (g) dengan parameter yang sama seperti yang dijelaskan di atas.

3. histogram\_merah: Menghitung histogram untuk saluran warna merah (r) dengan parameter yang sama seperti yang dijelaskan di atas.

4. histogram\_warna: Menghitung histogram untuk seluruh gambar warna (color\_image). Pada argumen pertama, daftar saluran warna adalah [0,1,2] yang mewakili saluran warna biru, hijau, dan merah. Parameter lainnya tetap sama seperti dijelaskan sebelumnya, yaitu jumlah bin histogram (256) dan rentang nilai piksel (0 hingga 255 untuk nilai warna dalam skala RGB) untuk setiap saluran warna.

Dengan menggunakan cv2.calcHist(), kode tersebut menghitung histogram untuk saluran warna biru, hijau, dan merah secara terpisah, serta histogram untuk seluruh gambar warna.

A screenshot of a computer

Description automatically generatedSetelah menjalankan kode ini, Anda akan memiliki empat histogram yang siap digunakan untuk menganalisis distribusi intensitas warna dalam gambar. Histogram ini memberikan informasi tentang sebaran intensitas warna dalam gambar, yang dapat digunakan untuk berbagai tujuan, seperti analisis warna, segmentasi objek, atau klasifikasi gambar.

Kode ini menghasilkan subplot baru yang menampilkan histogram untuk gambar asli serta histogram untuk saluran warna merah, hijau, dan biru. Setiap subplot menampilkan histogram menggunakan fungsi plt.hist() dari Matplotlib, dengan konfigurasi sebagai berikut:

1. plt.figure(figsize=(20, 5)): Membuat sebuah gambar (figure) dengan ukuran 20x5 inci.

2. plt.subplot(141): Membuat subplot pertama dengan ukuran 1 baris dan 4 kolom, dan memilih subplot pertama sebagai tempat untuk menampilkan histogram gambar asli.

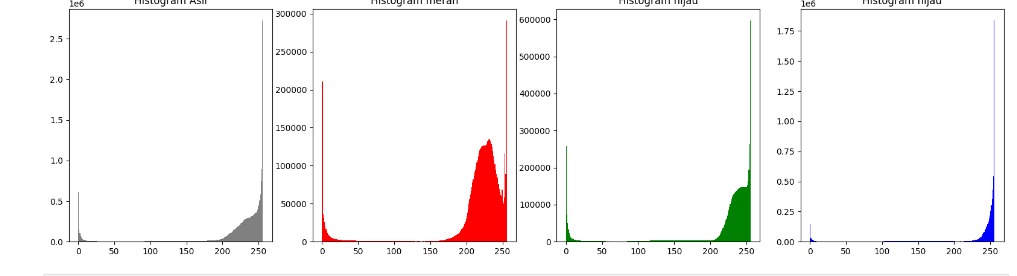
3. plt.hist(color\_image.flatten(), bins=256, range=[0,256], color='gray'): Menampilkan histogram gambar asli dengan nilai piksel yang sudah di-flatten, jumlah bin sebanyak 256, rentang nilai piksel dari 0 hingga 255, dan warna histogram adalah abu-abu.

4. plt.subplot(142), plt.subplot(143), plt.subplot(144): Membuat subplot untuk histogram saluran warna merah, hijau, dan biru dengan konfigurasi yang sama seperti di atas, namun dengan warna histogram yang sesuai dengan saluran warna masing-masing.

5. Pemanggilan plt.show() menampilkan semua subplot histogram yang telah dibuat.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Program ini melakukan beberapa operasi pemrosesan citra dan deteksi warna menggunakan OpenCV dan NumPy, kemudian menampilkan hasilnya menggunakan Matplotlib.

Berikut adalah rangkuman fungsionalitas setiap fungsi:

1. make\_image\_invisible(image): Menghasilkan gambar yang sama ukurannya dengan input tetapi dengan semua pikselnya berwarna hitam (0, 0, 0), sehingga gambar tersebut akan terlihat tidak terlihat.

2. increase\_brightness(image, value=30): Meningkatkan kecerahan gambar dengan nilai yang diberikan. Ini dilakukan dengan mengonversi gambar ke ruang warna HSV, meningkatkan komponen nilai (kecerahan), dan kemudian mengonversinya kembali ke ruang warna BGR.

3. detect\_and\_highlight\_blue(image): Mendeteksi warna biru dalam gambar dengan mengonversinya ke ruang warna HSV, membuat maska untuk warna biru, dan kemudian menggabungkan gambar asli dengan maska tersebut untuk menyorot warna biru.

4. detect\_and\_highlight\_red\_blue(image): Mendeteksi warna merah dan biru dalam gambar dengan mengonversinya ke ruang warna HSV, membuat maska untuk warna merah dan biru, dan kemudian menggabungkan gambar asli dengan kedua maska tersebut untuk menyorot warna merah dan biru.

5. detect\_and\_highlight\_red\_green\_blue(image): Mendeteksi warna merah, hijau, dan biru dalam gambar dengan mengonversinya ke ruang warna HSV, membuat maska untuk warna merah, hijau, dan biru, dan kemudian menggabungkan gambar asli dengan ketiga maska tersebut untuk menyorot warna merah, hijau, dan biru.

Setelah melakukan operasi pemrosesan citra, program menampilkan gambar asli, gambar yang tidak terlihat, serta hasil deteksi warna biru, merah-biru, dan merah-hijau-biru dalam subplot menggunakan Matplotlib.

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A collage of images of different colors

Description automatically generated