A white rectangular object with a black border

Description automatically generated

Program diatas adalah pengolahan gambar dan menampilkan gambar menggunakan matplotlib.Ini, seperti membuka gambar dengan OpenCV dan menampilkannya dengan Matplotlib.

* import cv2: Ini mengimpor modul cv2 (OpenCV) yang digunakan untuk pengolahan gambar dan video.
* import numpy as np: Ini mengimpor modul numpy dengan alias np. Numpy adalah pustaka Python yang digunakan untuk bekerja dengan array, vektor, dan matriks dalam Python. Biasanya digunakan dalam konteks pengolahan gambar untuk manipulasi array gambar.
* from matplotlib import pyplot as plt: Ini mengimpor fungsi pyplot dari modul matplotlib dengan alias plt. Matplotlib adalah pustaka Python yang digunakan untuk membuat visualisasi seperti grafik, plot, dan gambar. Dalam konteks ini, kita menggunakan pyplot untuk menampilkan gambar.
* Program ini adalah implementasi dasar untuk membaca dan menampilkan gambar menggunakan OpenCV dan Matplotlib di lingkungan Jupyter Notebook atau IPython. Mari kita jelaskan setiap barisnya:
* import cv2: Ini mengimpor modul cv2 (OpenCV) yang digunakan untuk pengolahan gambar dan video.
* import numpy as np: Ini mengimpor modul numpy dengan alias np. Numpy adalah pustaka Python yang digunakan untuk bekerja dengan array, vektor, dan matriks dalam Python. Biasanya digunakan dalam konteks pengolahan gambar untuk manipulasi array gambar.
* import matplotlib.pyplot as plt: Ini mengimpor fungsi pyplot dari modul matplotlib dengan alias plt. Matplotlib adalah pustaka Python yang digunakan untuk membuat visualisasi seperti grafik, plot, dan gambar. Dalam konteks ini, kita menggunakan pyplot untuk menampilkan gambar.
* import matplotlib.image as img: Ini mengimpor modul image dari matplotlib dengan alias img. Modul ini digunakan untuk membaca dan menampilkan gambar dalam Matplotlib.
* %matplotlib inline: Ini adalah perintah khusus untuk notebook Jupyter/IPython yang memastikan bahwa gambar yang dihasilkan oleh Matplotlib akan ditampilkan di dalam notebook itu sendiri, bukan dalam jendela popup terpisah.

Program ini hanya menyiapkan lingkungan untuk membaca, memproses, dan menampilkan gambar menggunakan OpenCV dan Matplotlib. Untuk menampilkan gambar, kita perlu menambahkan kode yang membaca gambar, misalnya menggunakan cv2.imread() dan kemudian menampilkannya menggunakan plt.imshow().

A screen shot of a computer

Description automatically generated

Kode ini membaca gambar dengan nama file 'nilm.jpeg' menggunakan fungsi img.imread() dari modul matplotlib.image dan menyimpannya dalam variabel color\_image. Kemudian, gambar tersebut ditampilkan menggunakan fungsi plt.imshow() dari Matplotlib.

Dengan senang hati! Mari kita jabarkan baris per baris:

* `color\_image = img.imread('nilm.jpeg')`: Baris ini membaca gambar dengan nama file 'nilm.jpeg' menggunakan fungsi `img.imread()` dari modul `matplotlib.image`. Hasilnya disimpan dalam variabel `color\_image`. Fungsi `img.imread()` mengembalikan gambar sebagai array NumPy.
* `plt.imshow(color\_image)`: Baris ini menampilkan gambar yang telah dibaca sebelumnya menggunakan fungsi `plt.imshow()` dari Matplotlib. Fungsi ini mengambil array gambar sebagai argumen dan menampilkannya sebagai gambar. Ini hanyalah representasi visual dari array gambar yang telah dibaca.

A close up of a sign

Description automatically generatedA white background with green text

Description automatically generated

Baris-baris kode ini digunakan untuk memisahkan kanal warna dari gambar asli dan menampilkannya bersama dengan gambar asli dalam beberapa subplot.

* `r = color\_image[:, :, 0]`: Ini adalah operasi slicing yang digunakan untuk memisahkan kanal merah dari gambar. Dalam representasi array NumPy, dimensi pertama adalah baris, dimensi kedua adalah kolom, dan dimensi ketiga adalah kanal warna. Dengan menggunakan `[:, :, 0]`, kita mengambil semua baris dan kolom dari kanal pertama, yang merupakan kanal merah.
* `g = color\_image[:, :, 1]`: Ini adalah operasi slicing yang digunakan untuk memisahkan kanal hijau dari gambar. Dengan menggunakan `[:, :, 1]`, kita mengambil semua baris dan kolom dari kanal kedua, yang merupakan kanal hijau.
* `b = color\_image[:, :, 2]`: Ini adalah operasi slicing yang digunakan untuk memisahkan kanal biru dari gambar. Dengan menggunakan `[:, :, 2]`, kita mengambil semua baris dan kolom dari kanal ketiga, yang merupakan kanal biru.
* `f, (c1, c2, c3, c4) = plt.subplots(1, 4, figsize = (20,10))`: Baris ini membuat subplot dengan 1 baris dan 4 kolom menggunakan `plt.subplots()`. Ukuran total subplot adalah 20x10 inch (figsize=(20,10)). Fungsi ini mengembalikan sebuah objek figur (`f`) dan array dari subplot (`c1`, `c2`, `c3`, `c4`).
* `c1.set\_title('Gambar Asli')`: Ini menetapkan judul 'Gambar Asli' ke subplot pertama (`c1`).
* `c1.imshow(color\_image)`: Ini menampilkan gambar asli di subplot pertama (`c1`) menggunakan `plt.imshow()`.
* `c1.axis('off')`: Ini menghilangkan sumbu (garis tepi) dari subplot pertama (`c1`), sehingga tidak ada sumbu yang ditampilkan.
* `c2.set\_title('Merah')`: Ini menetapkan judul 'Merah' ke subplot kedua (`c2`).
* `c2.imshow(r, cmap="gray")`: Ini menampilkan kanal merah di subplot kedua (`c2`) menggunakan `plt.imshow()` dengan colormap "gray". Colormap "gray" digunakan karena kanal warna yang dipisahkan adalah gambar grayscale.
* `c2.axis('off')`: Ini menghilangkan sumbu (garis tepi) dari subplot kedua (`c2`).
* `c3.set\_title('Hijau')`: Ini menetapkan judul 'Hijau' ke subplot ketiga (`c3`).
* `c3.imshow(g, cmap="gray")`: Ini menampilkan kanal hijau di subplot ketiga (`c3`) menggunakan `plt.imshow()` dengan colormap "gray".
* `c3.axis('off')`: Ini menghilangkan sumbu (garis tepi) dari subplot ketiga (`c3`).
* `c4.set\_title('Biru')`: Ini menetapkan judul 'Biru' ke subplot keempat (`c4`).
* `c4.imshow(b, cmap="gray")`: Ini menampilkan kanal biru di subplot keempat (`c4`) menggunakan `plt.imshow()` dengan colormap "gray".
* A screenshot of a computer

  Description automatically generated`c4.axis('off')`: Ini menghilangkan sumbu (garis tepi) dari subplot keempat (`c4`).

Dengan menggunakan subplot, kita dapat membandingkan gambar asli dengan setiap kanal warna yang dipisahkan (merah, hijau, biru) secara bersamaan dalam satu tampilan.  Baris kode ini menggunakan fungsi `cv2.cvtColor()` dari OpenCV untuk mengubah skema warna gambar

* `cv2.cvtColor()`: Ini adalah fungsi dalam pustaka OpenCV yang digunakan untuk mengubah skema warna gambar.
* `color\_image`: Ini adalah gambar yang ingin diubah skema warnanya. Dalam kasus ini, gambar tersebut sudah dibaca sebelumnya menggunakan `img.imread()` atau fungsi lainnya, dan disimpan dalam variabel `color\_image`.
* `cv2.COLOR\_BGR2RGB`: Ini adalah parameter yang menentukan jenis konversi warna yang ingin dilakukan. Dalam kasus ini, konversi dilakukan dari skema warna BGR (Blue-Green-Red) ke skema warna RGB (Red-Green-Blue).

OpenCV secara default membaca gambar dalam skema warna BGR, sementara Matplotlib mengharapkan gambar dalam skema warna RGB. Oleh karena itu, konversi ini diperlukan agar gambar dapat ditampilkan dengan benar oleh Matplotlib.

Jadi, baris kode ini mengubah skema warna gambar dari BGR ke RGB menggunakan fungsi `cv2.cvtColor()`. Setelah ini, gambar yang disimpan dalam `color\_image` akan dapat ditampilkan dengan benar menggunakan Matplotlib. Untuk menghitung histogram dari masing-masing kanal warna (biru, hijau, merah) serta histogram warna keseluruhan dari gambar.

* `histogram\_biru = cv2.calcHist([b], [0], None, [256], [0, 256])`: Ini adalah pemanggilan fungsi `cv2.calcHist()` dari OpenCV untuk menghitung histogram dari kanal biru (`b`) dari gambar. Parameter pertama adalah daftar gambar, dalam hal ini hanya ada satu gambar (kanal biru). Parameter kedua adalah indeks kanal dalam gambar (dalam hal ini, 0 untuk kanal biru). Parameter ketiga adalah mask, yang dalam hal ini adalah `None` yang menandakan tidak ada mask yang digunakan. Parameter keempat adalah jumlah bin histogram, di sini disetel ke 256. Parameter kelima adalah rentang nilai piksel, di sini dari 0 hingga 255 (0 hingga 255 untuk intensitas piksel).
* `histogram\_hijau = cv2.calcHist([g],[0], None,[256],[0,256])`: Ini adalah langkah serupa untuk menghitung histogram dari kanal hijau (`g`). Parameter yang disediakan sama seperti pada perhitungan histogram biru.
* `histogram\_merah = cv2.calcHist([r], [0], None,[256],[0,256])`: Ini adalah langkah serupa untuk menghitung histogram dari kanal merah (`r`). Parameter yang disediakan sama seperti pada perhitungan histogram biru.
* `histogram\_warna = cv2.calcHist([color\_image], [0,1,2],None, [256, 256, 256], [0, 256, 0, 256, 0, 256])`: Ini adalah langkah untuk menghitung histogram warna keseluruhan dari gambar. Dalam hal ini, kita memberikan daftar gambar yang berisi gambar asli (`color\_image`). Parameter kedua adalah daftar indeks kanal, yang dalam hal ini `[0,1,2]` mewakili kanal biru, hijau, dan merah. Parameter ketiga adalah mask, yang dalam hal ini `None` menandakan tidak ada mask yang digunakan. Parameter keempat adalah jumlah bin histogram untuk masing-masing kanal, di sini kita menggunakan `[256, 256, 256]` untuk masing-masing kanal. Parameter kelima adalah rentang nilai piksel untuk masing-masing kanal, di sini dari 0 hingga 255 untuk masing-masing kanal (0 hingga 255 untuk intensitas piksel).

Dengan menggunakan `cv2.calcHist()`, kita dapat menghasilkan histogram untuk masing-masing kanal warna serta histogram warna keseluruhan dari gambar. Histogram ini berguna untuk menganalisis distribusi intensitas piksel dalam gambar.

A screenshot of a computer

Description automatically generated

Untuk menampilkan histogram dari masing-masing kanal warna (biru, hijau, merah) serta histogram warna keseluruhan dari gambar dalam subplot yang berbeda-beda. Program ini menggunakan library `matplotlib` untuk membuat sebuah figure dengan ukuran (20, 5). Kemudian, program ini membagi figure tersebut menjadi empat subplot menggunakan `plt.subplot`.

* Subplot pertama (`plt.subplot(141)`) menampilkan histogram dari seluruh piksel dalam gambar warna. Histogram ini menunjukkan distribusi intensitas warna dari keseluruhan gambar. Histogram ini diplot dengan judul "Histogram Asli" dan warna abu-abu.
* Subplot kedua (`plt.subplot(142)`) menampilkan histogram dari saluran warna merah (red) dalam gambar. Histogram ini menunjukkan distribusi intensitas warna merah dalam gambar. Histogram ini diplot dengan judul "Histogram Merah" dan warna merah.
* Subplot ketiga (`plt.subplot(143)`) menampilkan histogram dari saluran warna hijau (green) dalam gambar. Histogram ini menunjukkan distribusi intensitas warna hijau dalam gambar. Histogram ini diplot dengan judul "Histogram Hijau" dan warna hijau.
* Subplot keempat (`plt.subplot(144)`) menampilkan histogram dari saluran warna biru (blue) dalam gambar. Histogram ini menunjukkan distribusi intensitas warna biru dalam gambar. Histogram ini diplot dengan judul "Histogram Biru" dan warna biru.

Setelah mengatur setiap subplot, program menggunakan `plt.show()` untuk menampilkan keseluruhan figure beserta keempat subplotnya. Dengan demikian, kita bisa melihat histogram dari gambar asli serta histogram dari masing-masing saluran warna (merah, hijau, dan biru) secara bersamaan untuk memahami distribusi intensitas warna dalam gambar tersebut.

A screenshot of a computer

Description automatically generated

Program penggunaan OpenCV untuk melakukan beberapa operasi pada gambar.

* + \*\*`make\_image\_invisible(image)`\*\*:
    - Fungsi ini mengambil gambar sebagai input dan mengembalikan gambar dengan semua pikselnya diatur menjadi nol, sehingga gambar tersebut menjadi "tidak terlihat" (invisible). Ini dilakukan dengan membuat matriks nol dengan dimensi yang sama dengan gambar input.
  + \*\*`increase\_brightness(image, value=30)`\*\*:
    - Fungsi ini mengambil gambar sebagai input dan secara opsional nilai peningkatan kecerahan (`value`) yang defaultnya adalah 30.
    - Pertama, gambar dikonversi ke ruang warna HSV (Hue, Saturation, Value).
    - Nilai kecerahan (value) dalam ruang warna HSV diubah dengan menambahkan nilai yang ditentukan (`value`).
    - Setelah itu, nilai kecerahan dibatasi agar tetap dalam rentang 0-255 (nilai minimum dan maksimum yang dapat direpresentasikan dalam gambar).
    - Hasil HSV yang sudah diubah kemudian dikonversi kembali ke ruang warna BGR (Blue, Green, Red) yang digunakan oleh OpenCV, dan hasilnya dikembalikan.
  + \*\*`detect\_and\_highlight\_blue(image)`\*\*:
    - Fungsi ini mengambil gambar sebagai input dan mengembalikan gambar dengan hanya bagian yang berwarna biru yang ditonjolkan.
    - Gambar dikonversi ke ruang warna HSV.
    - Dilakukan thresholding pada gambar dalam ruang warna HSV untuk mendeteksi warna biru, dan masker biru dihasilkan.
    - A screenshot of a computer code

      Description automatically generatedHasil masker digunakan untuk melakukan operasi bitwise and pada gambar asli, sehingga hanya piksel yang sesuai dengan warna biru yang akan ditampilkan.
  + \*\*`detect\_and\_highlight\_red\_blue(image)`\*\*:
    - Fungsi ini mirip dengan `detect\_and\_highlight\_blue`, tetapi juga memperhitungkan warna merah.
    - Dilakukan thresholding untuk mendeteksi warna merah dalam dua rentang (karena rentang warna merah melintasi nol dalam ruang warna HSV), serta untuk warna biru.
    - Masker hasil dari kedua warna digabungkan menggunakan operasi bitwise or.
    - Hasil masker digunakan untuk menyorot warna merah dan biru pada gambar asli.
  + \*\*`detect\_and\_highlight\_red\_green\_blue(image)`\*\*:
    - Fungsi ini serupa dengan sebelumnya, tetapi juga memperhitungkan warna hijau.
    - Thresholding dilakukan untuk mendeteksi warna merah, hijau, dan biru masing-masing.
    - Masker hasil dari ketiga warna tersebut digabungkan menggunakan operasi bitwise or.
    - Hasil masker digunakan untuk menyorot warna merah, hijau, dan biru pada gambar asli.

Setelah mendefinisikan semua fungsi, program mengambil gambar dari file "namasaya.jpg" menggunakan OpenCV. Kemudian, gambar tersebut diolah menggunakan setiap fungsi yang telah didefinisikan sebelumnya. Hasilnya ditampilkan dalam sebuah figure matplotlib dengan empat subplot, masing-masing menampilkan gambar hasil dari setiap operasi.

