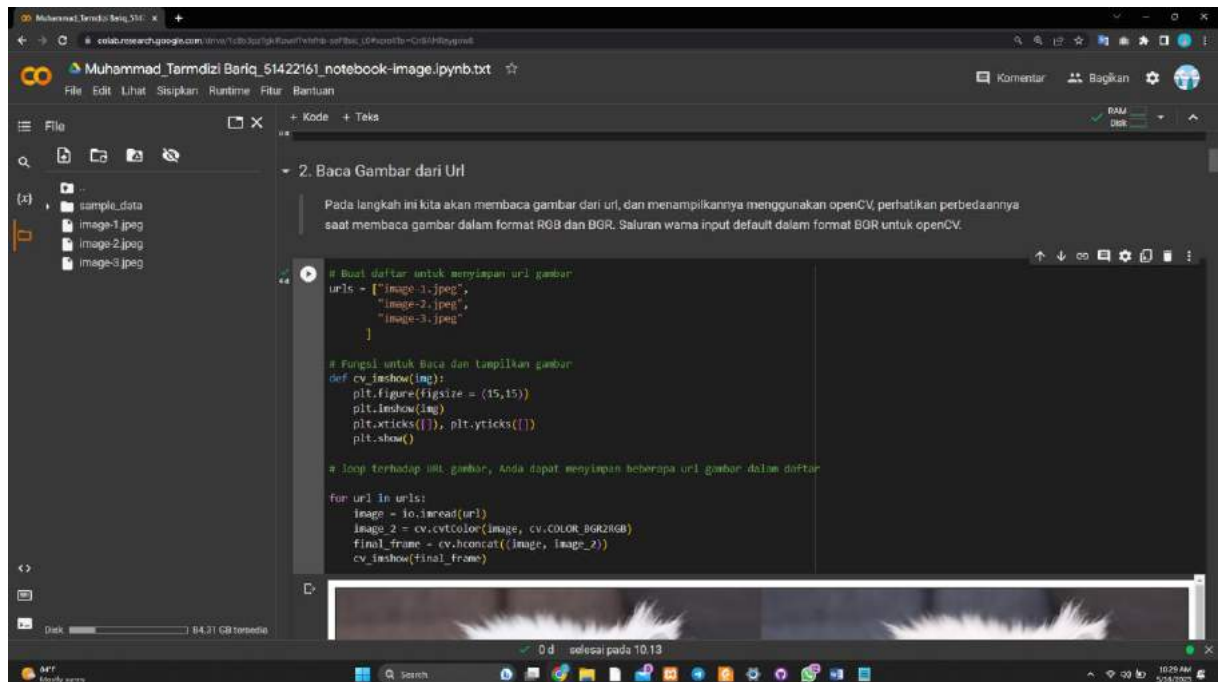
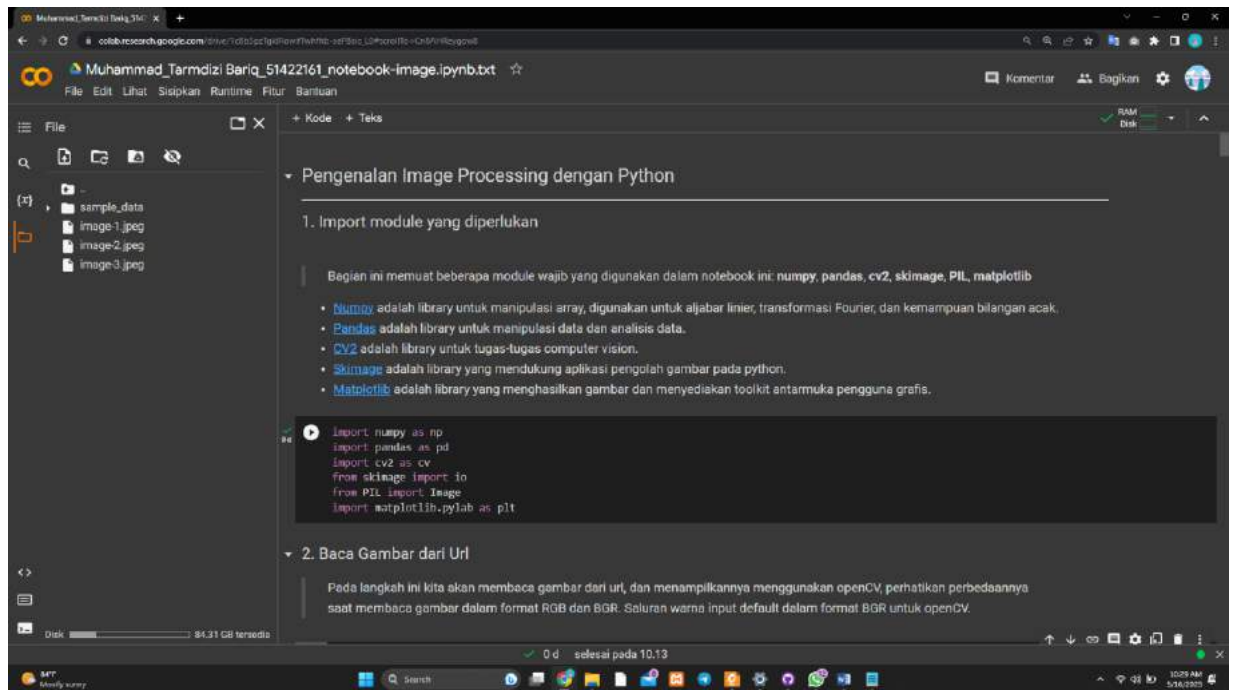
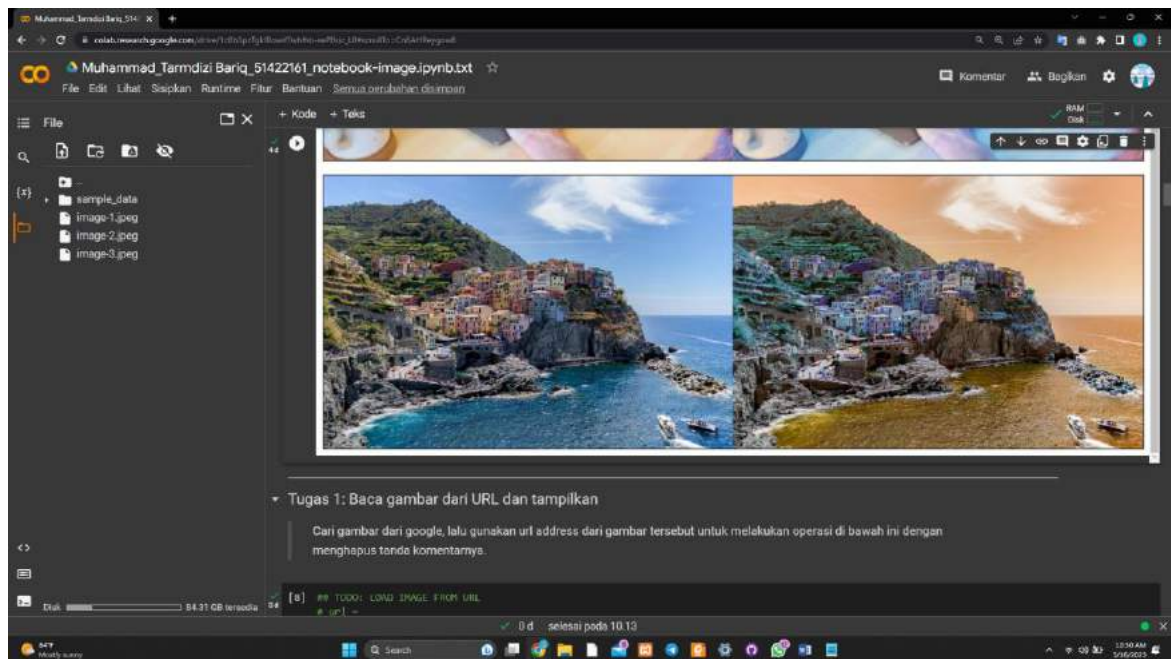
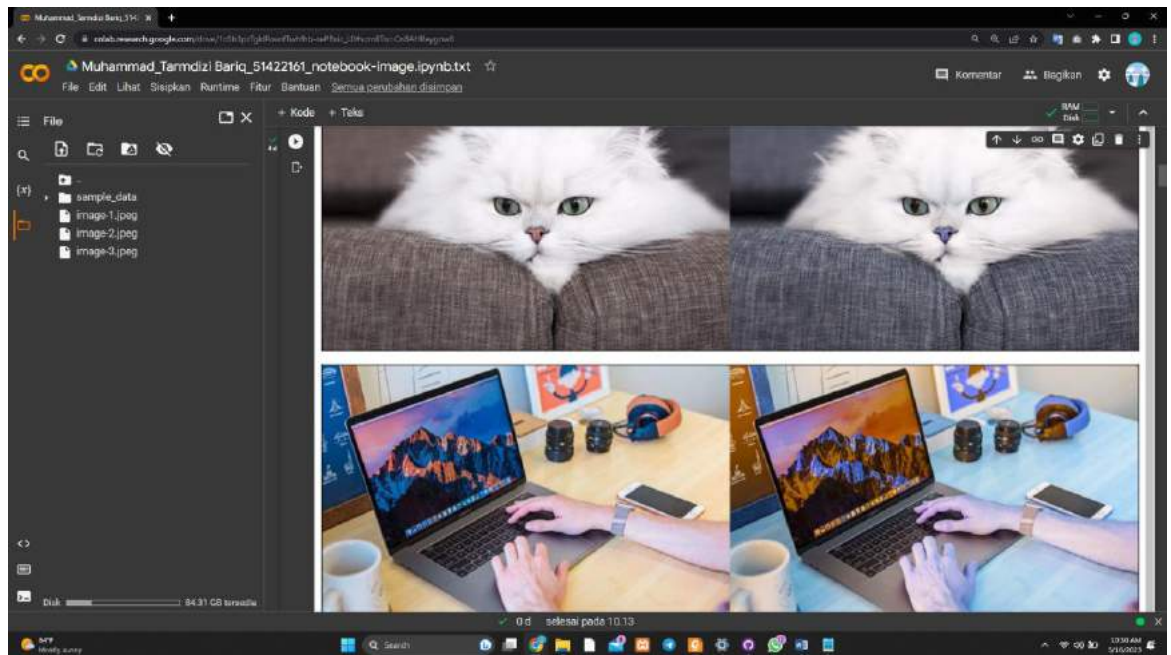


TUGAS PRAKTIKUM PERTEMUAN 8

NAMA : MUHAMMAD TARMIDZI BARIQ
KELAS : IIA13
NPM : 51422161





Muhammad_Tarmidzi Bariq_51422161_notebook-image.ipynb.txt

File Edit Lihat Simpan Runtime Fitur Bantuan Semua perubahan disimpan

File

- sample_data
 - image-1.jpeg
 - image-2.jpeg
 - image-3.jpeg

Tugas 1: Baca gambar dari URL dan tampilkan

Cari gambar dari google, lalu gunakan url address dari gambar tersebut untuk melakukan operasi di bawah ini dengan menghapus tanda komentarnya.

```
[8] # TODO: LOAD IMAGE FROM URL
# url =
# myimg = io.imread(url)
# cv.imshow(cv.cvtColor(myimg, cv.COLOR_BGR2RGB))
```

3. Kontur Gambar dan Histogram

```
[9] # Gunakan image dari indeks pertama dalam variabel uris
image = io.imread(uris[0])
```

Menghasilkan Histogram citra berwarna dan citra grayscale

Terkadang Anda ingin meningkatkan kontras pada gambar atau memperluas kontras di wilayah tertentu sambil mengorbankan detail dalam warna yang tidak terlalu bervariasi, atau tidak penting. Alat yang baik untuk menemukan wilayah yang menarik adalah histogram. Untuk membuat histogram dari data gambar kita, kita menggunakan fungsi `matplotlib.pyplot.hist()`.

Menampilkan histogram dari semua piksel pada gambar berwarna :

```
[10] plt.hist(image.ravel(), bins = 256, range = [0,256])
plt.show()
```

0 d selesai pada 10:13

Muhammad_Tarmidzi Bariq_51422161_notebook-image.ipynb.txt

File Edit Lihat Simpan Runtime Fitur Bantuan Semua perubahan disimpan

File

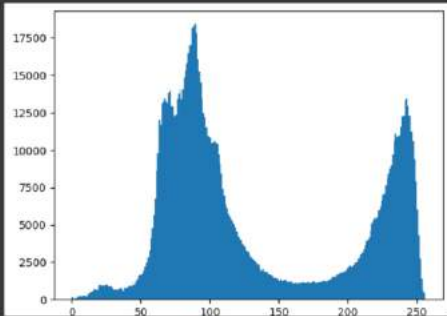
- sample_data
 - image-1.jpeg
 - image-2.jpeg
 - image-3.jpeg

Menghasilkan Histogram citra berwarna dan citra grayscale

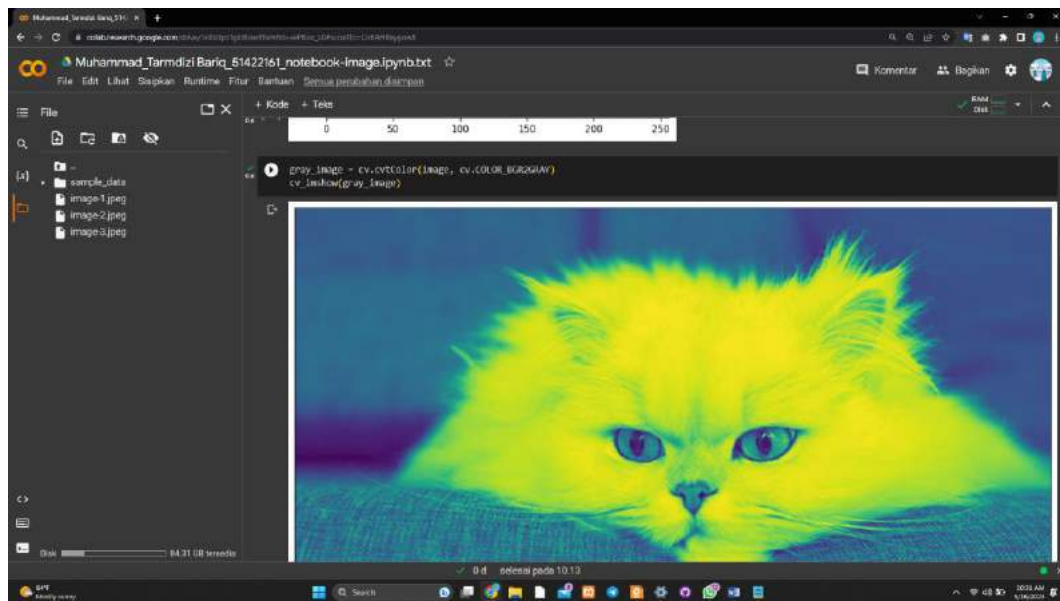
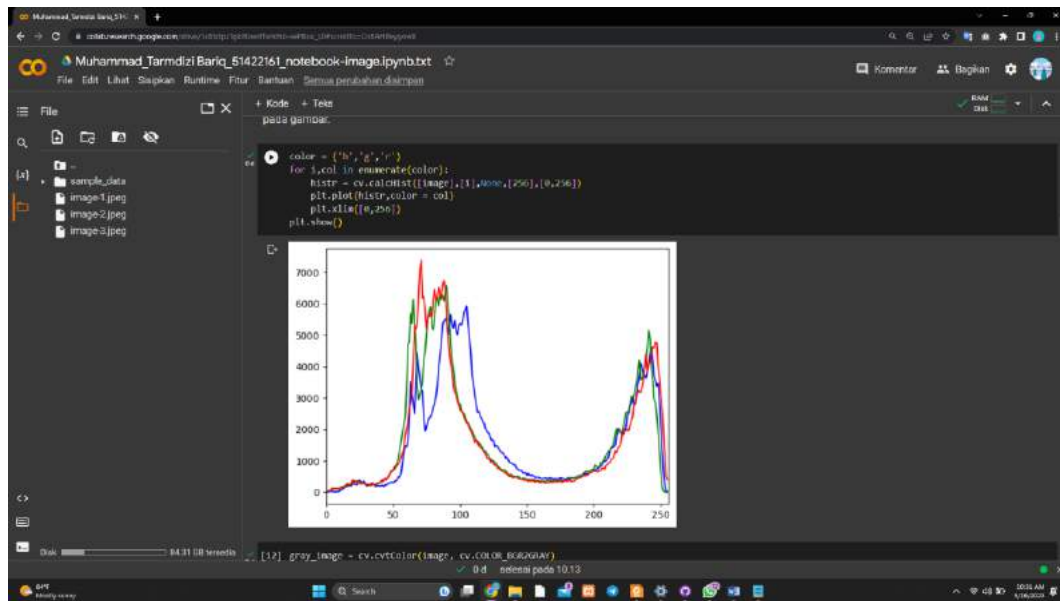
Terkadang Anda ingin meningkatkan kontras pada gambar atau memperluas kontras di wilayah tertentu sambil mengorbankan detail dalam warna yang tidak terlalu bervariasi, atau tidak penting. Alat yang baik untuk menemukan wilayah yang menarik adalah histogram. Untuk membuat histogram dari data gambar kita, kita menggunakan fungsi `matplotlib.pyplot.hist()`.

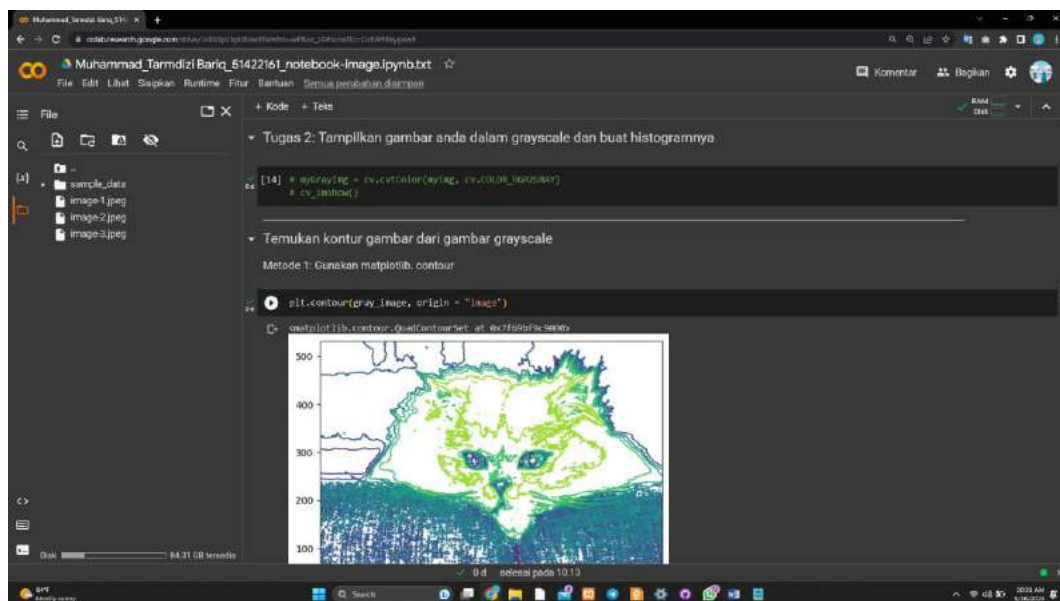
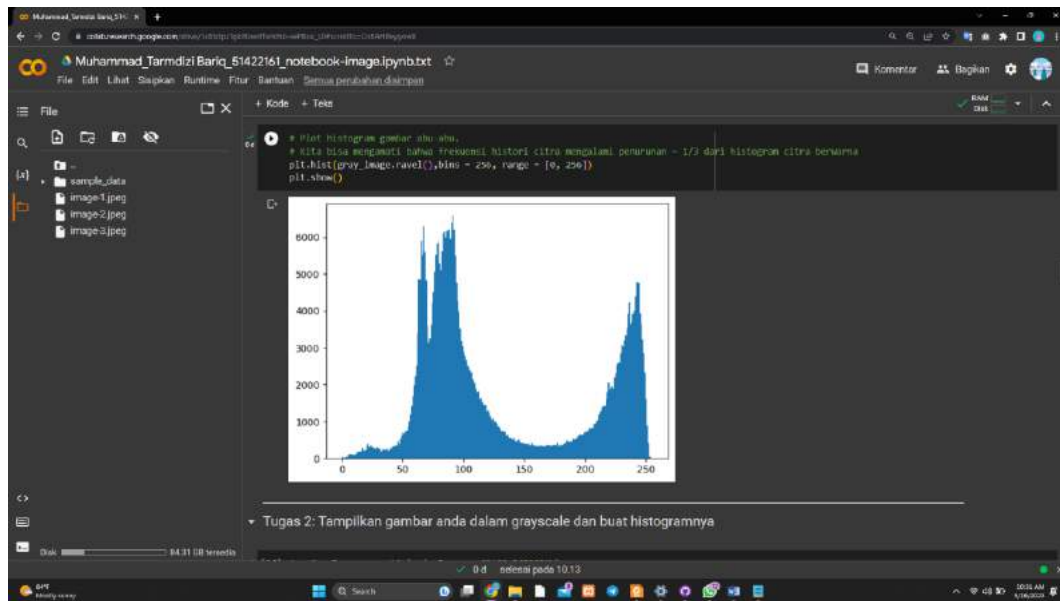
Menampilkan histogram dari semua piksel pada gambar berwarna :

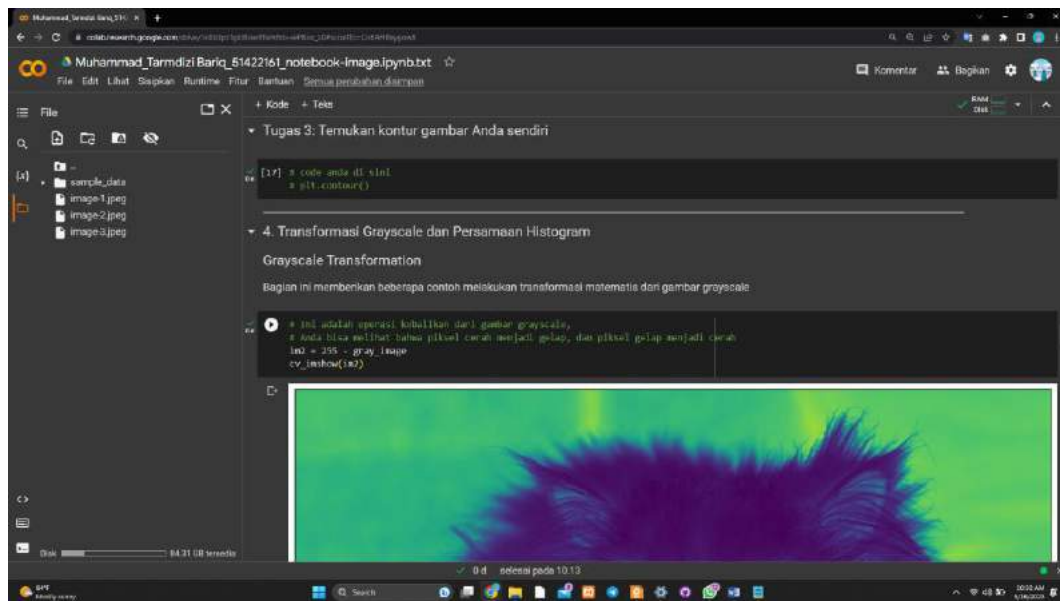
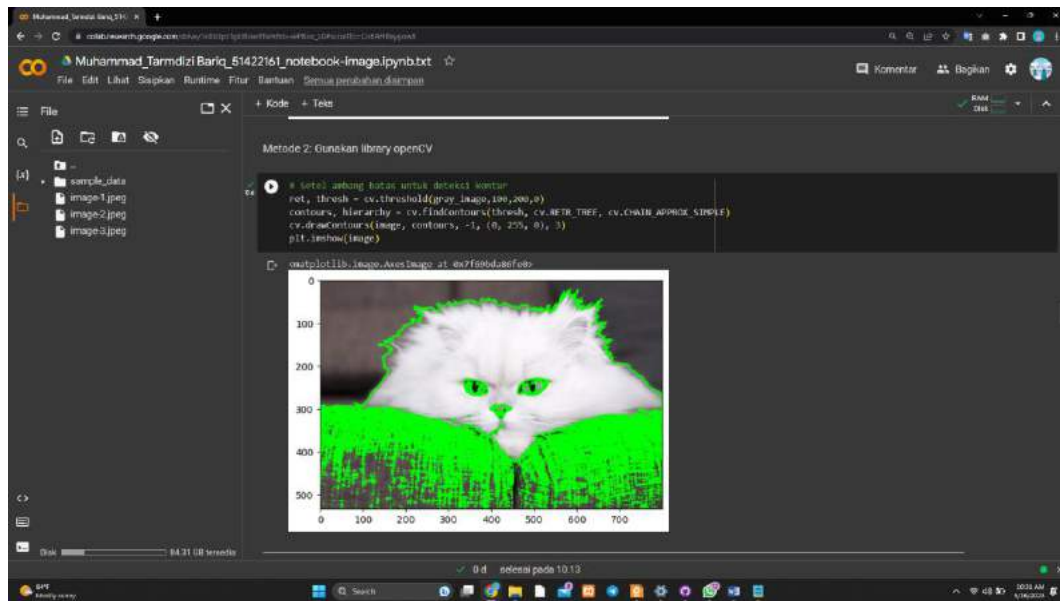
```
plt.hist(image.ravel(), bins = 256, range = [0,256])
plt.show()
```

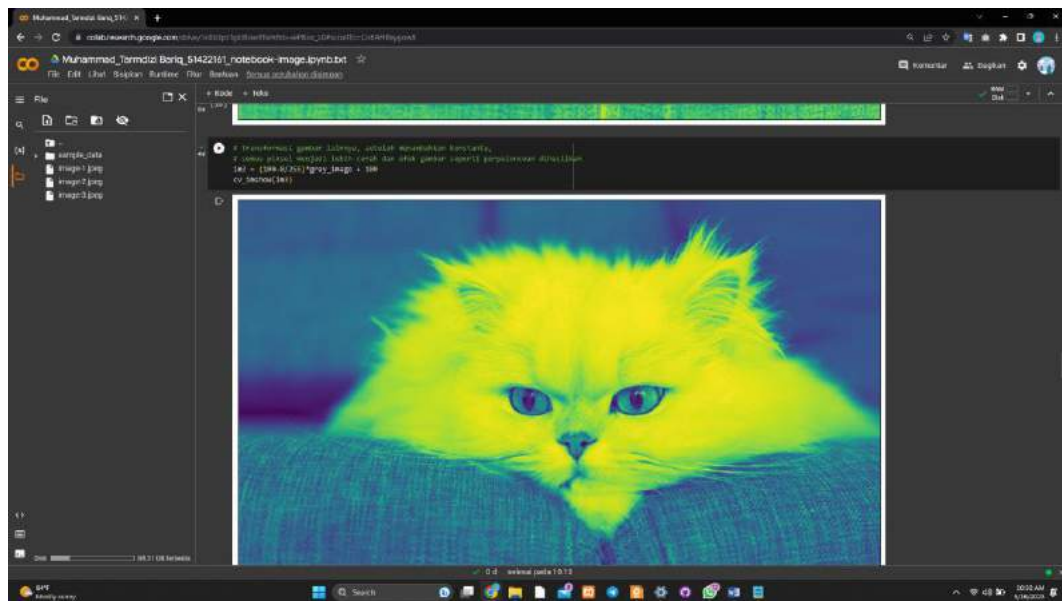
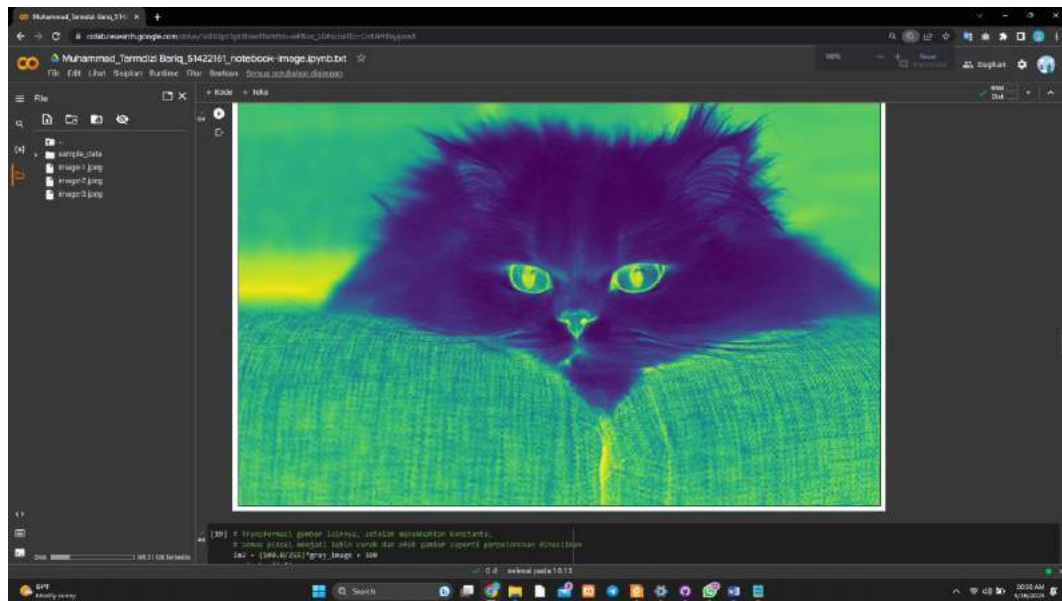


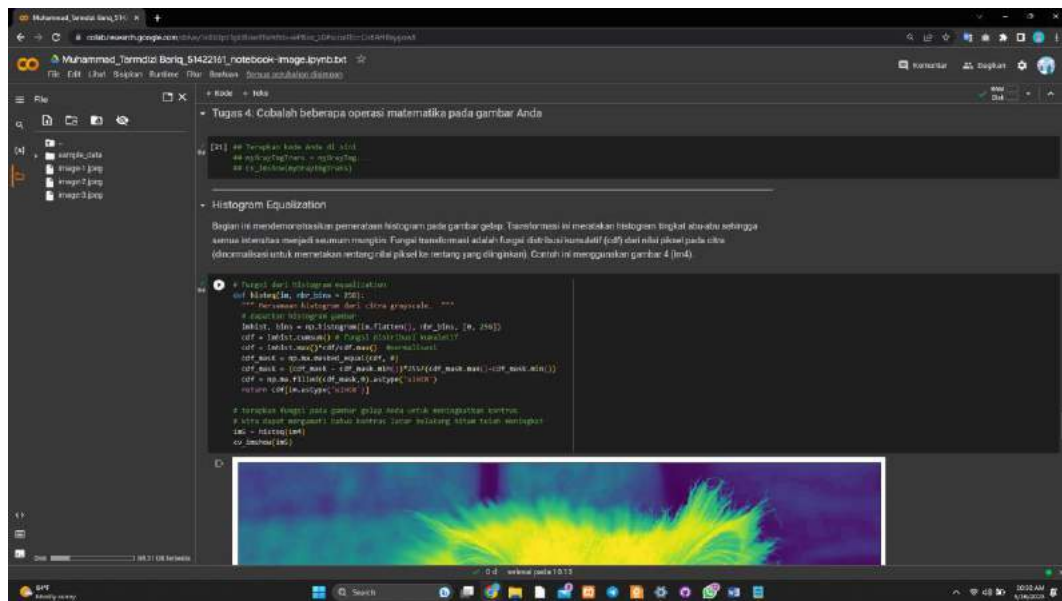
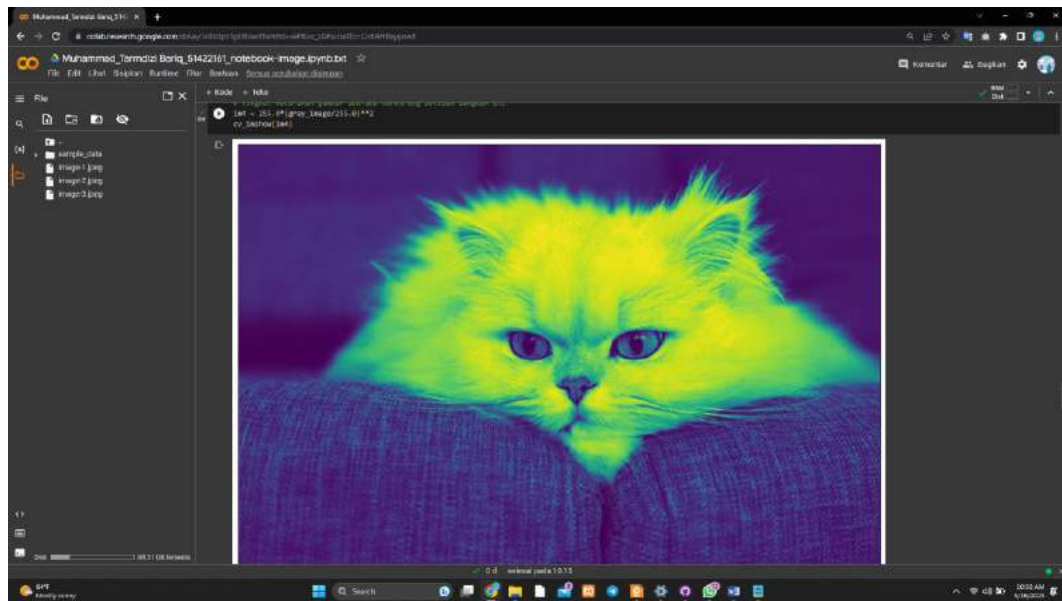
0 d selesai pada 10:13

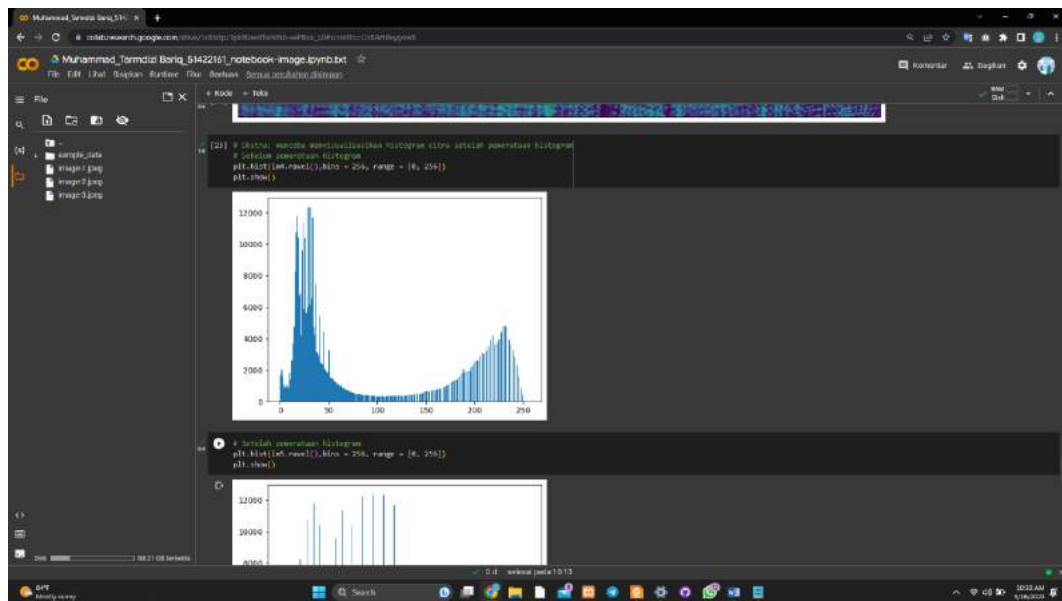
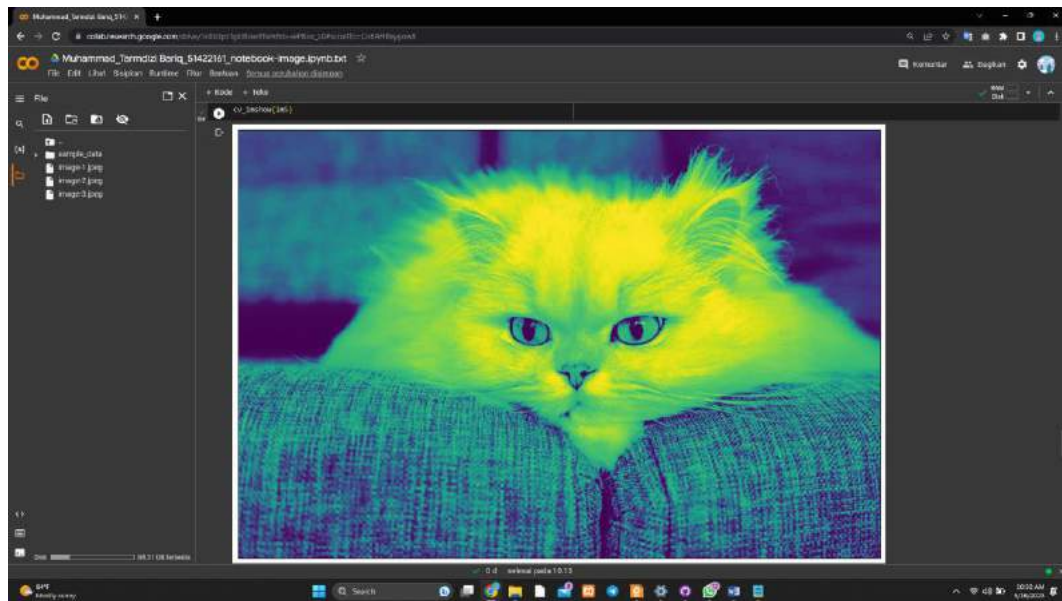


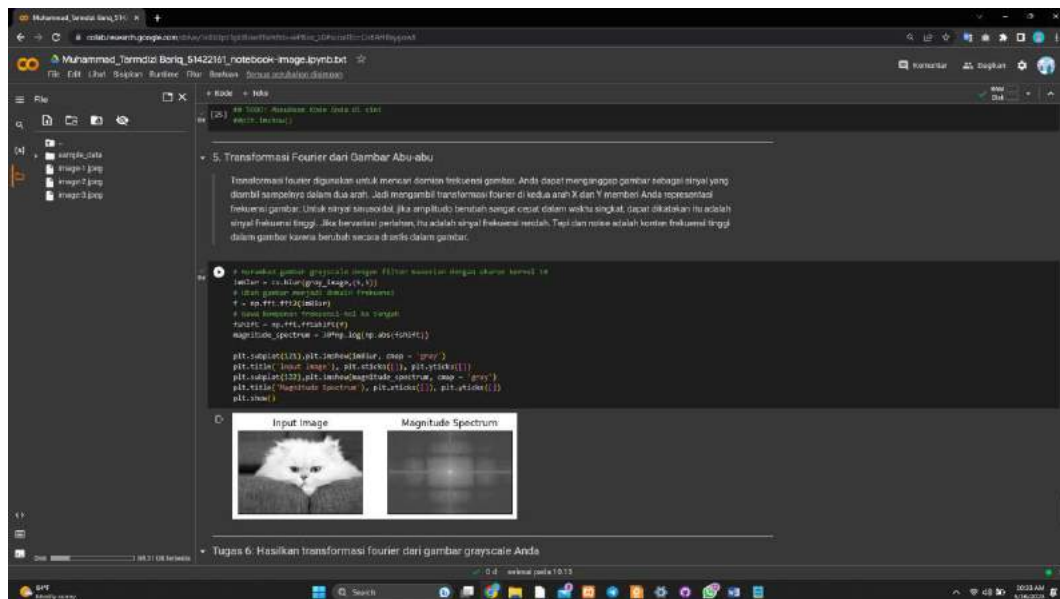
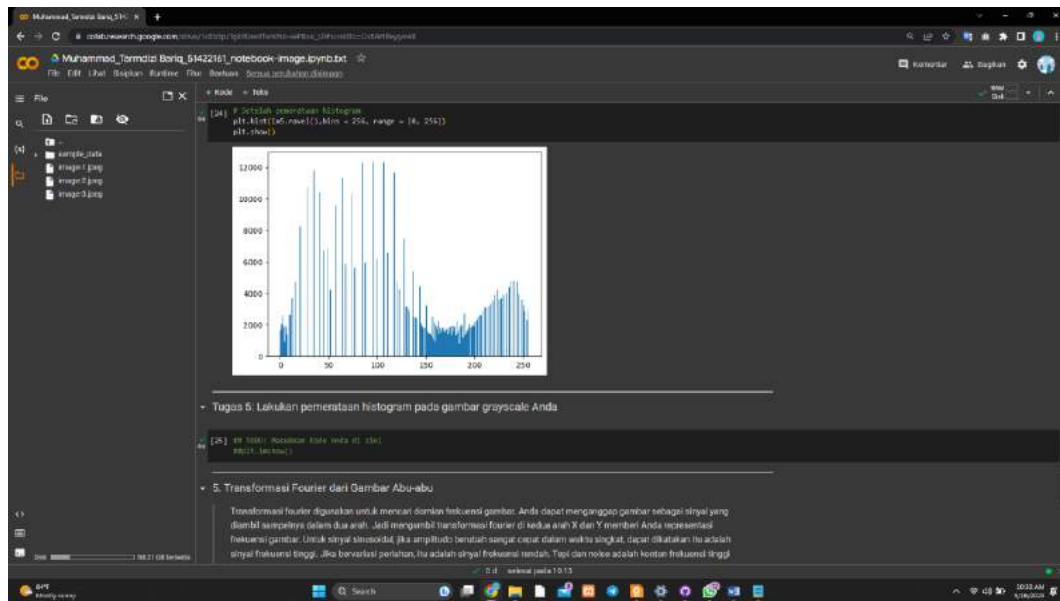


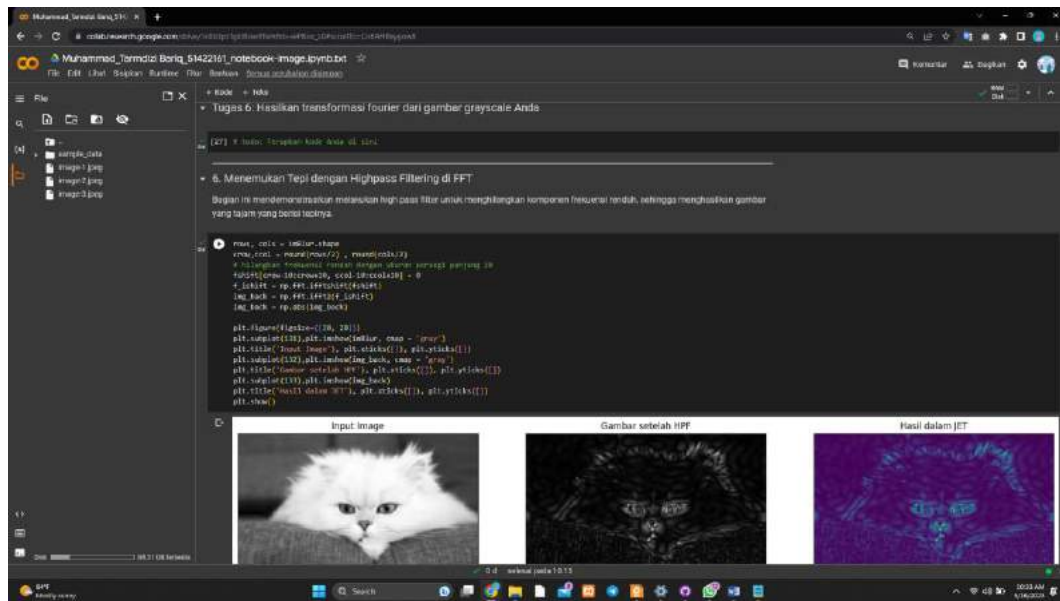












PENJELASAN :

Kode di atas adalah contoh implementasi pengolahan gambar menggunakan library seperti NumPy, pandas, OpenCV (cv2), scikit-image (skimage), PIL (Python Imaging Library), dan matplotlib.

- **Import Library:** Mengimpor library yang diperlukan untuk pengolahan gambar, seperti NumPy (np), pandas (pd), OpenCV (cv2), scikit-image (skimage), PIL (Image), dan matplotlib (plt).
- **Deklarasi URL Gambar:** Membuat daftar (urls) yang berisi URL gambar yang akan digunakan dalam contoh ini.
- **Fungsi cv_imshow():** Membuat fungsi cv_imshow() untuk menampilkan gambar menggunakan matplotlib.
- **Looping URL Gambar:** Melakukan loop terhadap setiap URL gambar yang ada dalam daftar (urls). Membaca gambar menggunakan skimage.io.imread(). Mengubah mode warna gambar dari BGR ke RGB menggunakan cv.cvtColor(). Menggabungkan dua gambar secara horizontal menggunakan cv.hconcat(). Menampilkan gambar yang sudah digabungkan menggunakan fungsi cv_imshow().
- **Looping URL Gambar:** Melakukan loop terhadap setiap URL gambar yang ada dalam daftar (urls). Membaca gambar menggunakan skimage.io.imread(). Mengubah mode warna gambar dari BGR ke RGB menggunakan cv.cvtColor(). Menggabungkan dua gambar secara horizontal menggunakan cv.hconcat(). Menampilkan gambar yang sudah digabungkan menggunakan fungsi cv_imshow().
- **Konversi ke Grayscale:** Mengubah gambar menjadi grayscale menggunakan cv.cvtColor(). Menampilkan gambar grayscale menggunakan cv_imshow(). Membuat histogram dari gambar grayscale menggunakan plt.hist(). Menampilkan histogram grayscale menggunakan plt.show().

- Kontur dan Pengolahan Tambahan: Membuat kontur dari gambar grayscale menggunakan `cv.threshold()` dan `cv.findContours()`. Menggambar kontur pada gambar asli menggunakan `cv.drawContours()`. Menampilkan gambar dengan kontur menggunakan `plt.imshow()`. Melakukan beberapa operasi pengolahan tambahan pada gambar grayscale, seperti inversi warna dan peningkatan kontras. Menampilkan gambar-gambar hasil pengolahan tambahan menggunakan `cv.imshow()`.
- Histogram Equalization: Membuat fungsi `histeq()` untuk melakukan equalisasi histogram pada gambar grayscale. Menggunakan fungsi `histeq()` untuk meningkatkan kontras gambar hasil pengolahan sebelumnya. Menampilkan gambar hasil equalisasi histogram menggunakan `cv.imshow()`. Menampilkan histogram gambar sebelum dan setelah equalisasi menggunakan `plt.hist()` dan `plt.show()`.
- Filter Gaussian dan Transformasi Fourier: Mengaburkan gambar grayscale menggunakan filter Gaussian dengan `cv.blur()`. Mengubah gambar ke domain frekuensi menggunakan transformasi Fourier dengan `np.fft.fft2()`. Menggeser komponen frekuensi-nol ke tengah dengan `np.fft.fftshift()`. Menghitung magnitudo spektrum dengan `30*np.log(np.abs(fshift))`. Menampilkan gambar asli yang sudah di-blur dan magnitudo spektrum menggunakan `plt.subplot()` dan `plt.imshow()`.