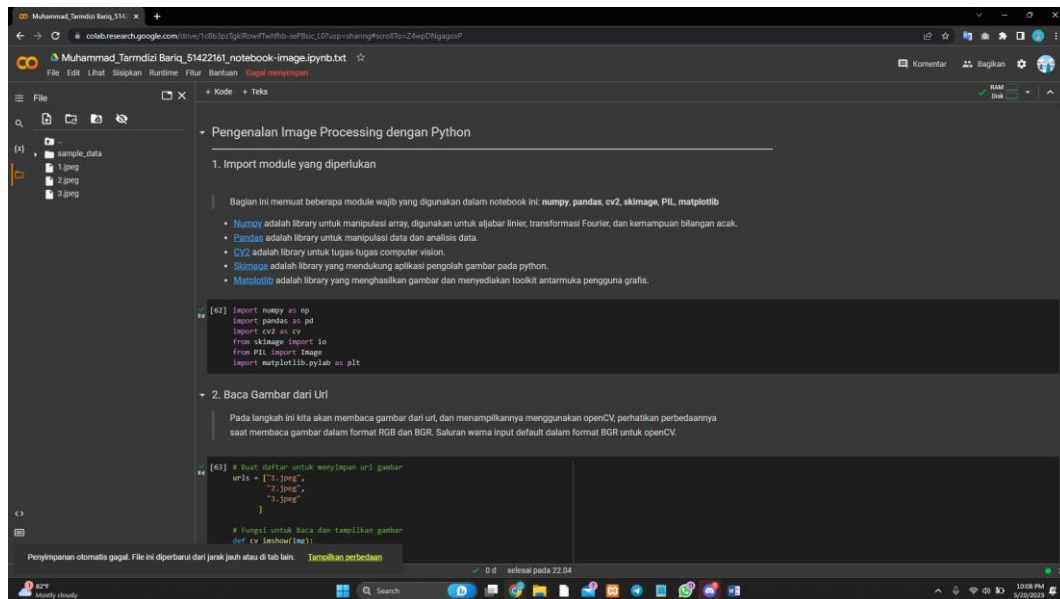


## TUGAS PRAKTIKUM PERTEMUAN 8

NAMA : MUHAMMAD TARMIDZI BARIQ  
KELAS : IIA13  
NPM : 51422161



```
1. Import module yang diperlukan

Bagian ini memuat beberapa module wajib yang digunakan dalam notebook ini: numpy, pandas, cv2, skimage, PIL, matplotlib

• Numpy adalah library untuk manipulasi array, digunakan untuk aljabar linier, transformasi Fourier, dan kemampuan bilangan acak.
• Pandas adalah library untuk manipulasi data dan analisis data.
• CV2 adalah library untuk tugas-tugas computer vision.
• Skimage adalah library yang mendukung aplikasi pengolahan gambar pada python.
• Matplotlib adalah library yang menghasilkan gambar dan menyediakan toolkit antarmuka pengguna grafis.

[42] import numpy as np
import pandas as pd
import cv2 as cv
from skimage import io
from PIL import Image
import matplotlib.pyplot as plt

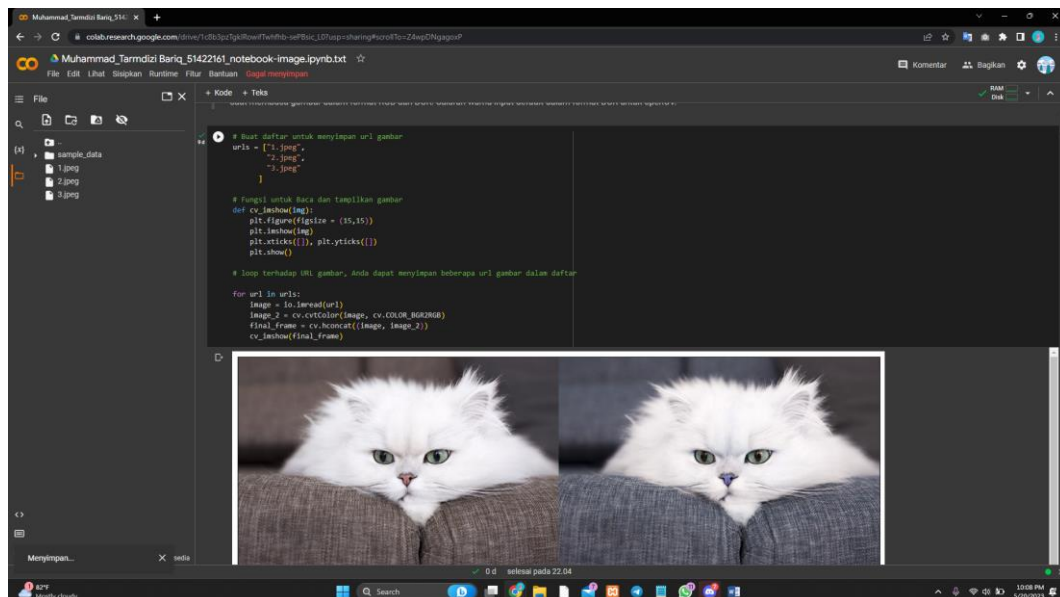
2. Baca Gambar dari Uri

Pada langkah ini kita akan membaca gambar dari uri, dan menampilkannya menggunakan openCV, perhatikan perbedaannya
saat membaca gambar dalam format RGB dan BGR. Saluran warna input default dalam format BGR untuk openCV.

[43] # Buat daftar untuk menyimpan uri gambar
uris = ["1.jpg",
        "2.jpg",
        "3.jpg"]


# Fungsi untuk Baca dan tampilkan gambar
def cv_show(img):
    print(img.shape)
    cv.imshow('img', img)

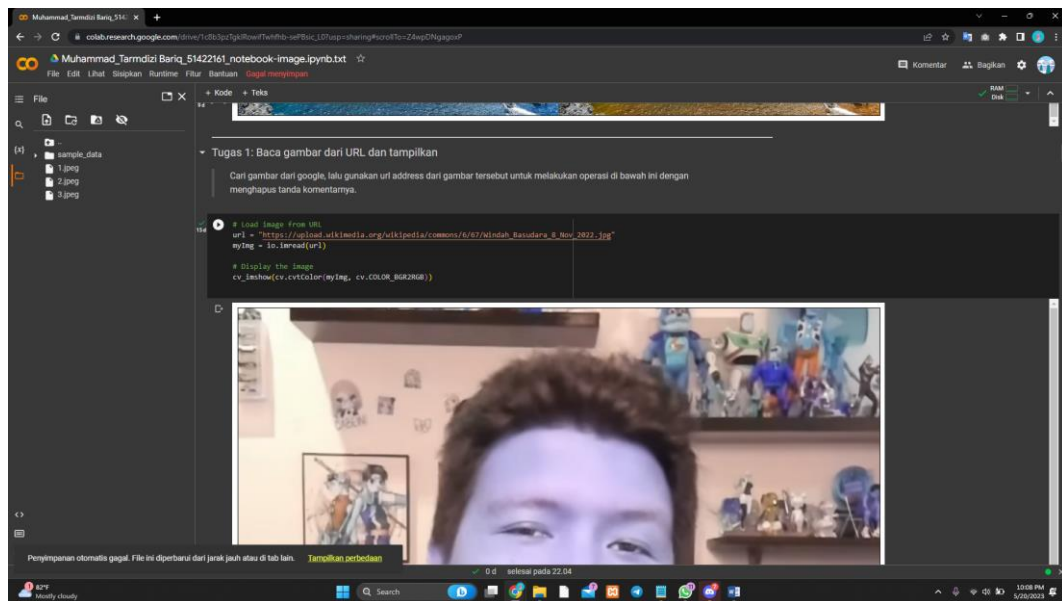
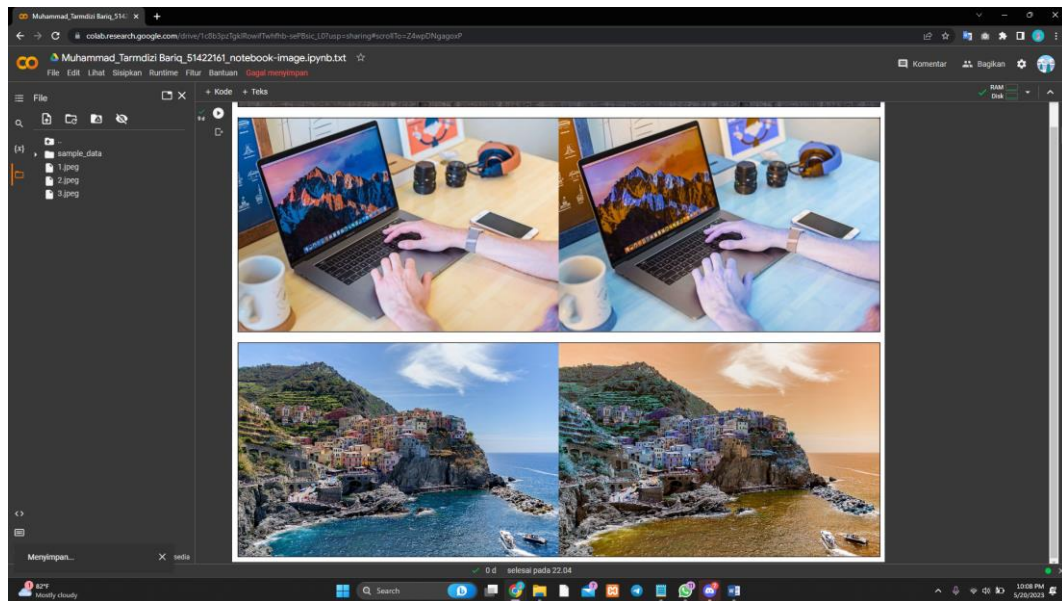
Penyimpanan otomatis gagal. File ini diperbarui dari jarak jauh atau di tab lain. Tampilkan perbedaan
```



```
# Fungsi untuk Baca dan tampilkan gambar
def cv_show(img):
    print(img.shape)
    cv.imshow('img', img)

# Loop terhadap uri gambar, Anda dapat menyimpan beberapa uri gambar dalam daftar
for uri in uris:
    image = io.imread(uri)
    image_2 = cv.cvtColor(image, cv.COLOR_BGR2RGB)
    final_frame = cv.hconcat([image, image_2])
    cv.imshow('final_frame', final_frame)
```





Muhammad, Tarmidzi Bariq, 514: x

colab.research.google.com/drive/1c8b3pTgkRowTwhFbb-wFbWc\_1D7up=sharing?scrollTo=Z4apDNagapP

File Edit Lihat Simpan Runtime Fitur Bantuan **Salah menyimpan**

Komentar Bagikan

RAM Disk


File

sample\_data  
1.png  
2.png  
3.png

+

Kode + Teks

[54]



3. Kontur Gambar dan Histogram

[55] # Simpan image dari index pertama dalam variabel url  
image = io.imread(url[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 :

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

Menyimpan... X

selesai pada 22.04

12° Mostly cloudy

13:08 PM 5/28/2024

Muhammad, Tarmidzi Bariq, 514: x

colab.research.google.com/drive/1c8b3pTgkRowTwhFbb-wFbWc\_1D7up=sharing?scrollTo=Z4apDNagapP

File Edit Lihat Simpan Runtime Fitur Bantuan **Salah menyimpan**

Komentar Bagikan

RAM Disk

File

sample\_data  
1.png  
2.png  
3.png

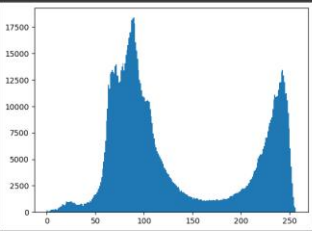
+

Kode + Teks

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 :

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



Menampilkan histogram saluran R, G, B Kita dapat mengamati bahwa saluran hijau memiliki banyak piksel di 255, yang mewakili tambahan putih pada gambar.

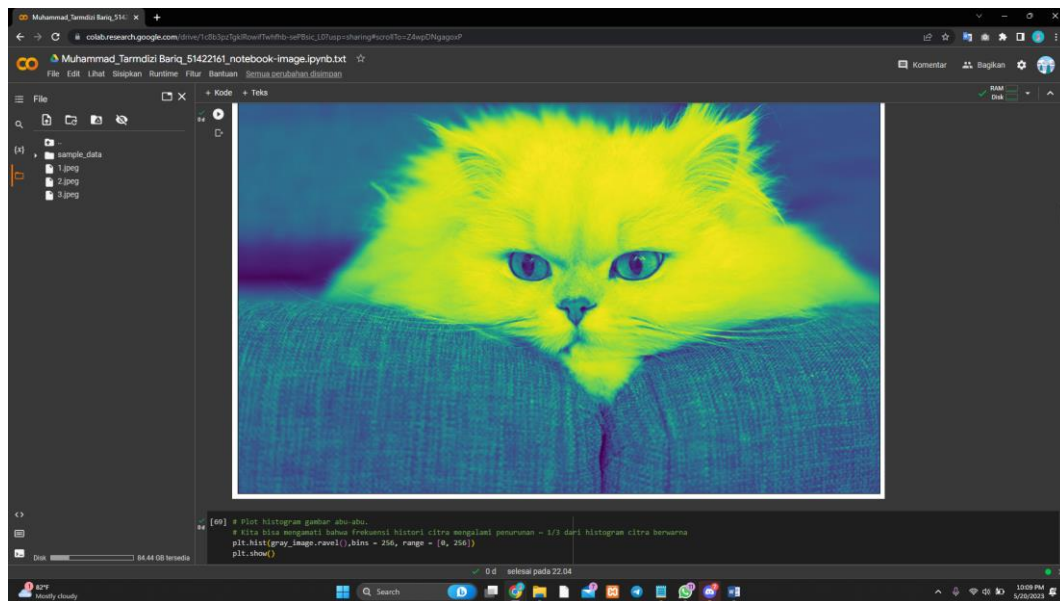
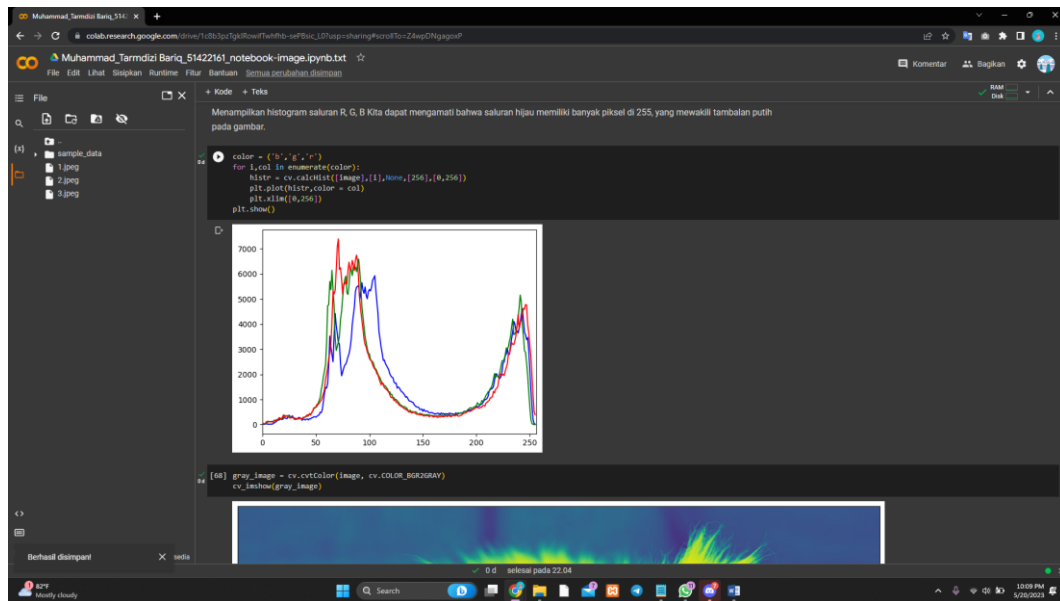
[57] color = ('b', 'g', 'r')  
for i, col in enumerate(color):

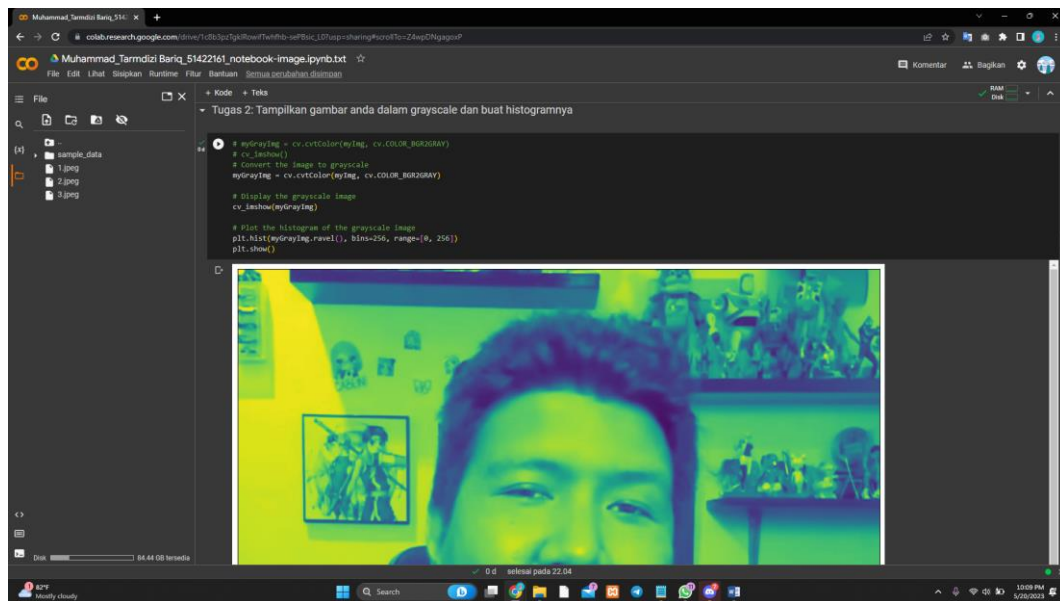
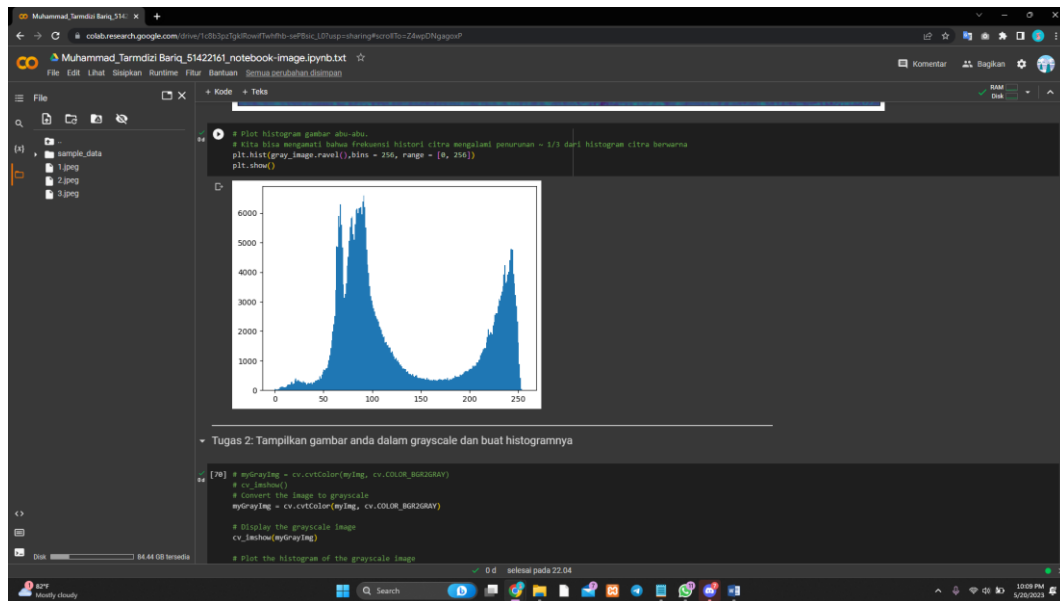
Menyimpan... X

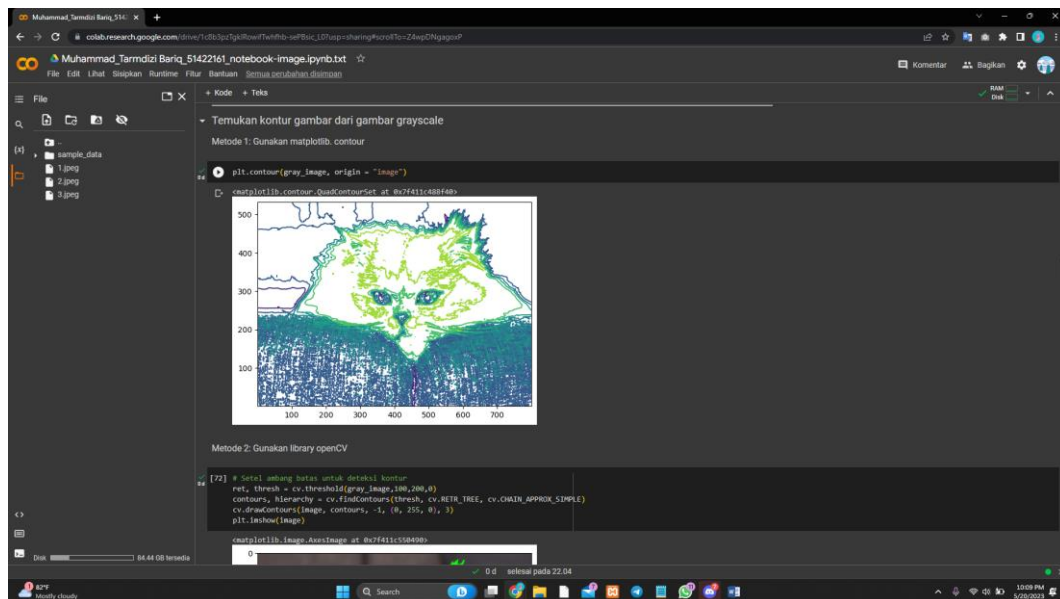
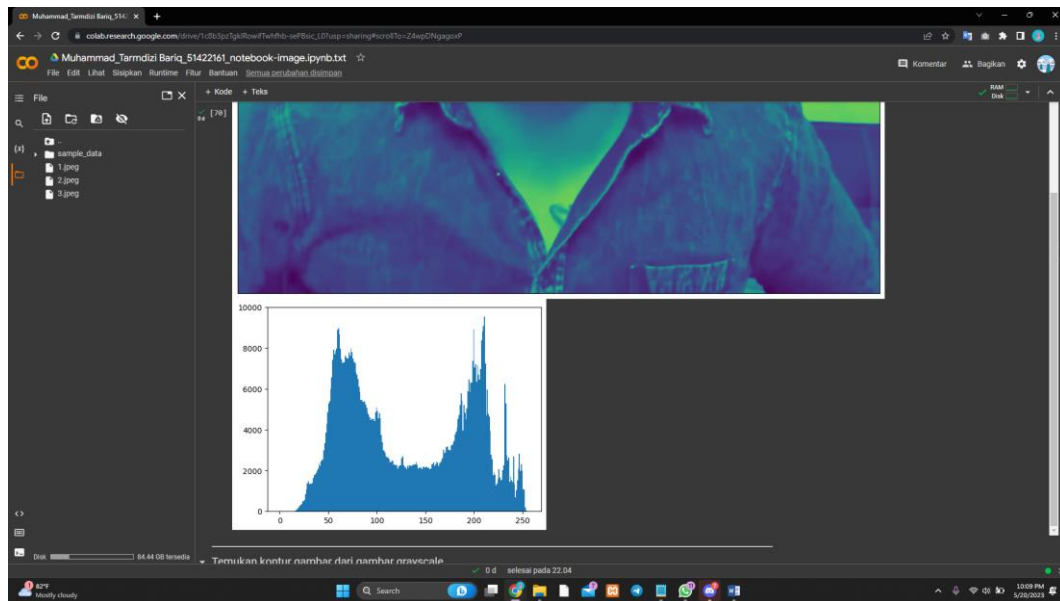
selesai pada 22.04

12° Mostly cloudy

13:08 PM 5/28/2024









Muhammad\_Tamridzi\_Bariq\_514: x

colab.research.google.com/drive/1d83j3gkRoweTufHb-saF8ac\_LDTup=sharing?scrollTo=Z4apDNagapsP

Muhammad\_Tamridzi\_Bariq\_51422161\_notebook-image.ipynb.txt

File Edit Lihat Simpan Runtime Filter Bantuan Tentang perubahan tampilan

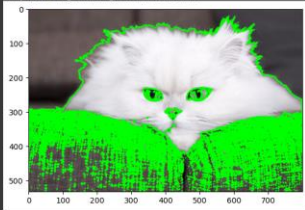
File

- 
- sample\_data
- 1.png
- 2.png
- 3.png

+

Kode + Teks


```
contours, hierarchy = cv.findContours(thresh, cv.RETR_TREE, cv.CHAIN_APPROX_SIMPLE)
cv.drawContours(image, contours, -1, (0, 255, 0), 3)
plt.imshow(image)
```



contour[0][1].image.AxesImage at 8x7f411:538498

Tugas 3: Temukan kontur gambar Anda sendiri

```
[73]: # code anda di sini
# plt.contour()
plt.contour(myGrayscale, origin = "image")
contour[0][1].contour.QuaKontourSet at 8x7f411:489088
```



Disk 84.44 GB tersedia

12°C Mostly cloudy

Search

13:08 PM 5/28/2024

Muhammad\_Tamridzi\_Bariq\_514: x

colab.research.google.com/drive/1d83j3gkRoweTufHb-saF8ac\_LDTup=sharing?scrollTo=Z4apDNagapsP

Muhammad\_Tamridzi\_Bariq\_51422161\_notebook-image.ipynb.txt

File Edit Lihat Simpan Runtime Filter Bantuan Tentang perubahan tampilan

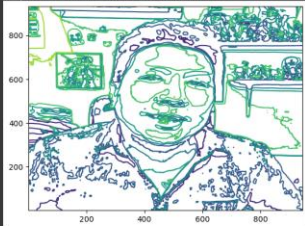
File

- 
- sample\_data
- 1.png
- 2.png
- 3.png

+

Kode + Teks

```
[73]: # plt.contour()
plt.contour(myGrayscale, origin = "image")
contour[0][1].contour.QuaKontourSet at 8x7f411:489088
```




4. Transformasi Grayscale dan Persamaan Histogram

Grayscale Transformation

Bagian ini memberikan beberapa contoh melakukan transformasi matematis dari gambar grayscale

```
[74]: # Ini adalah operasi kebalikan dari gambar grayscale,
# Anda bisa melihat bahwa piksel cerah menjadi gelap, dan piksel gelap menjadi cerah
img = 255 - gray_image
cv.imshow(img)
```



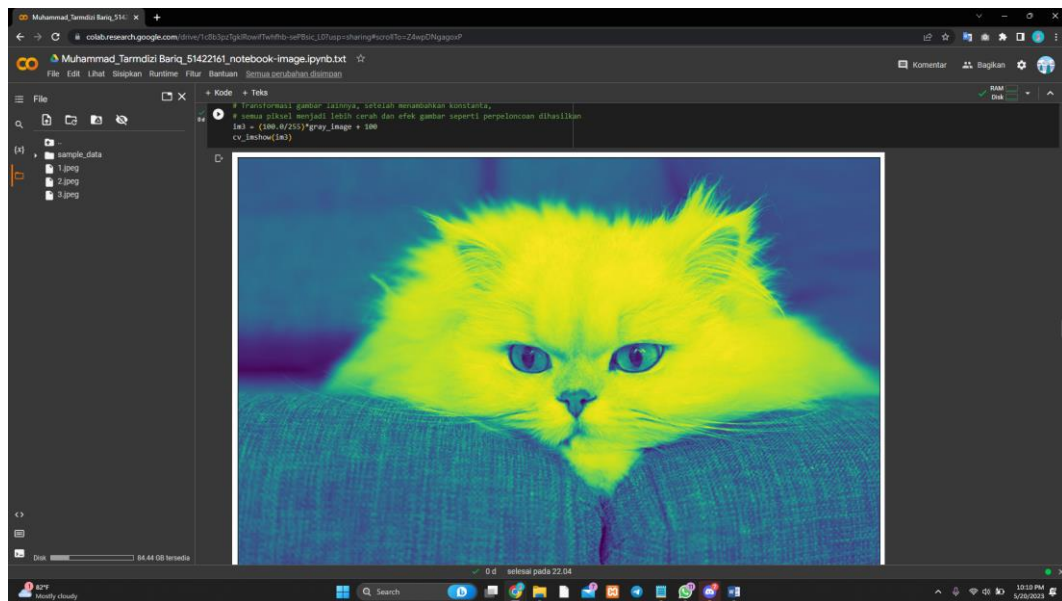
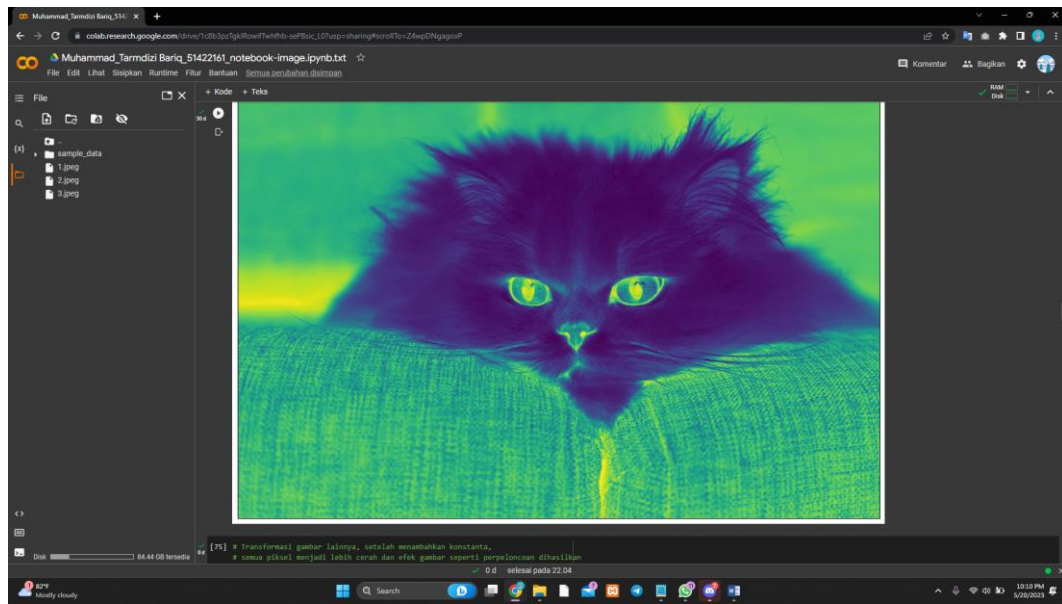
Disk 84.44 GB tersedia

12°C Mostly cloudy

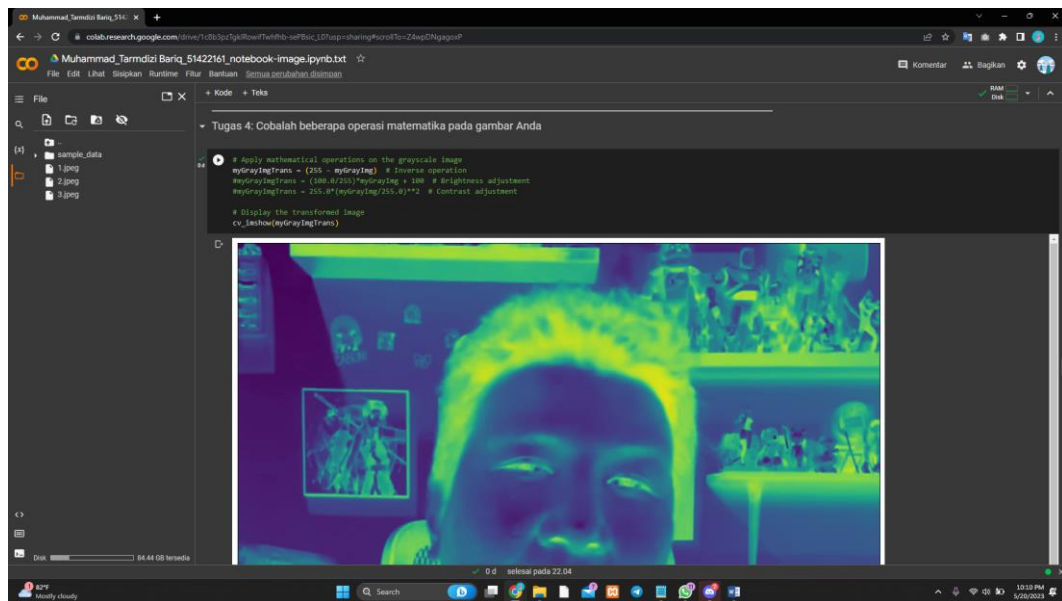
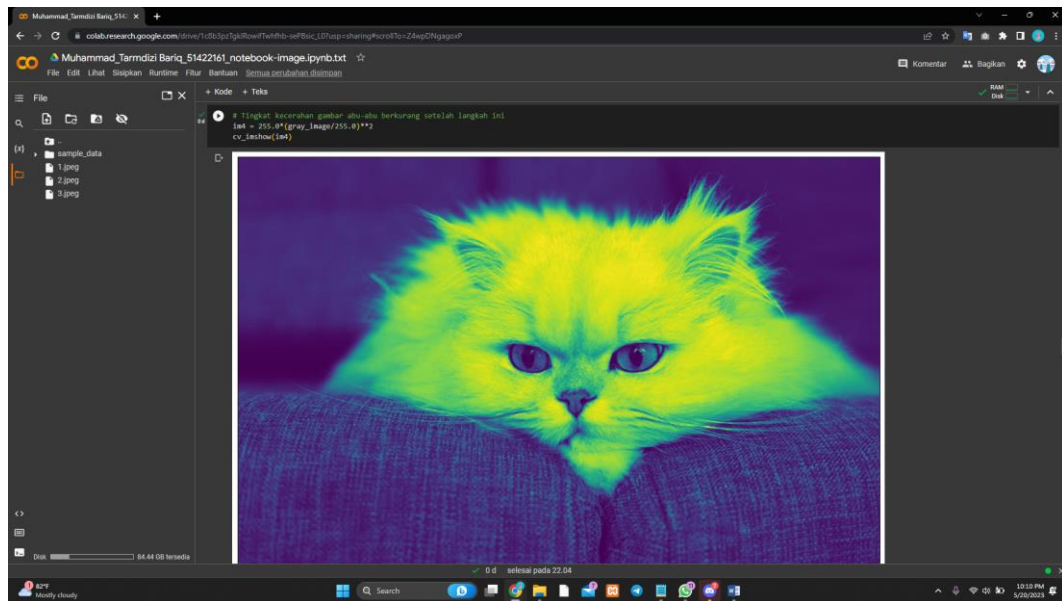
Search

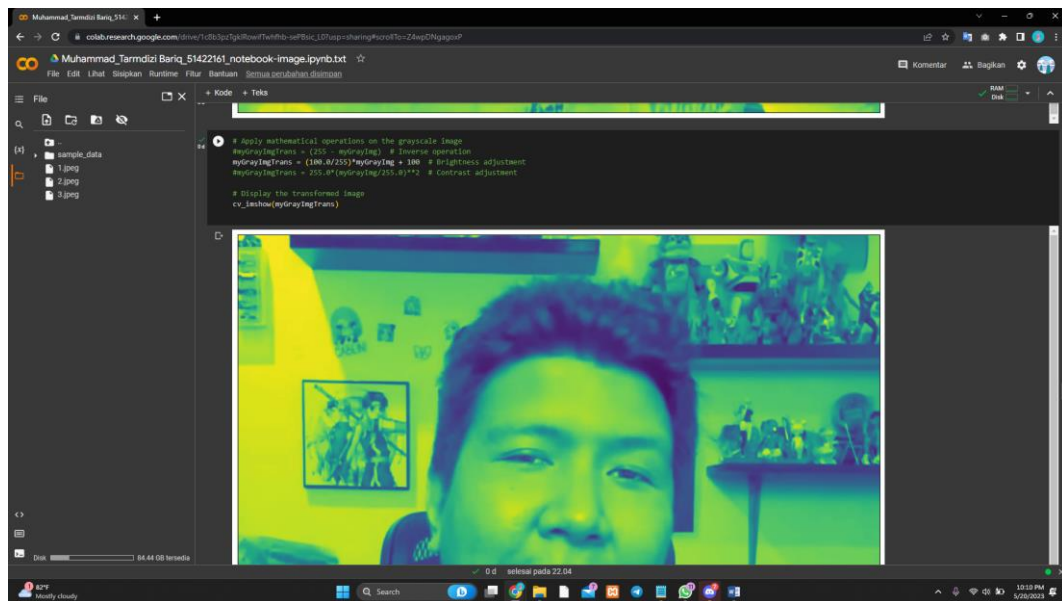
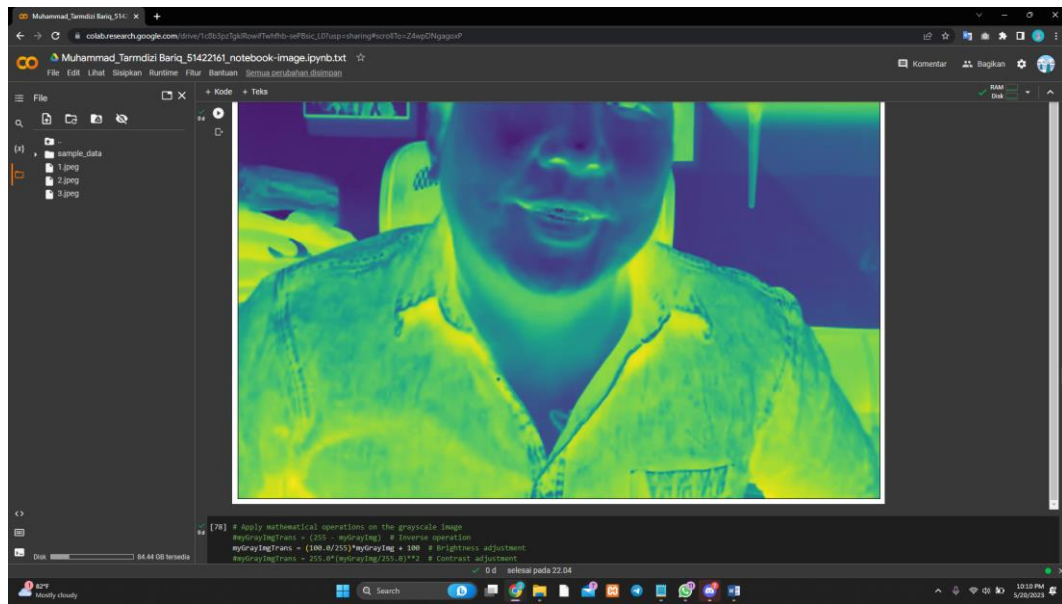
selesai pada 22:04

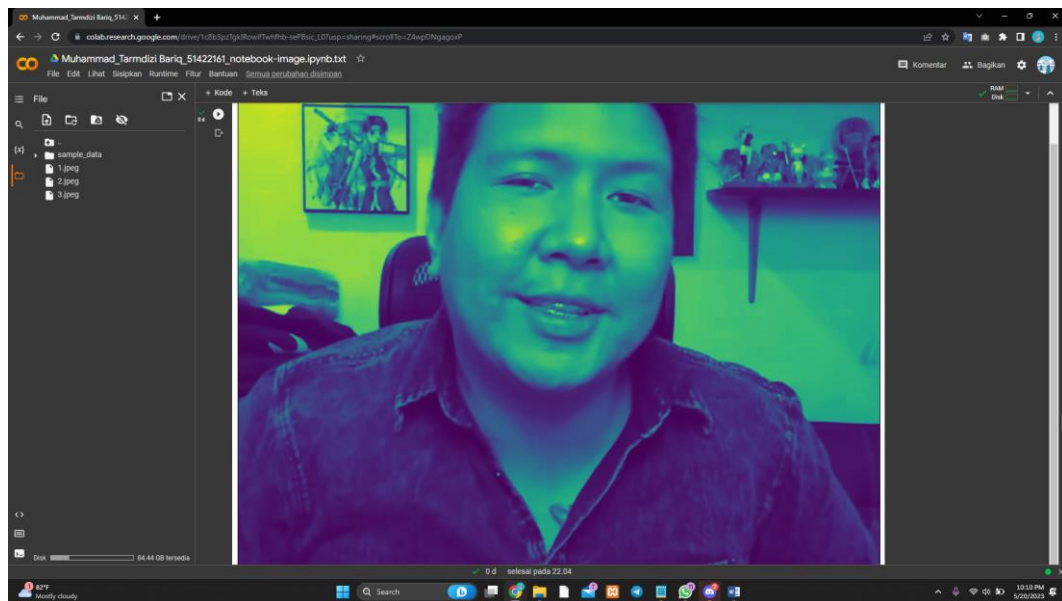
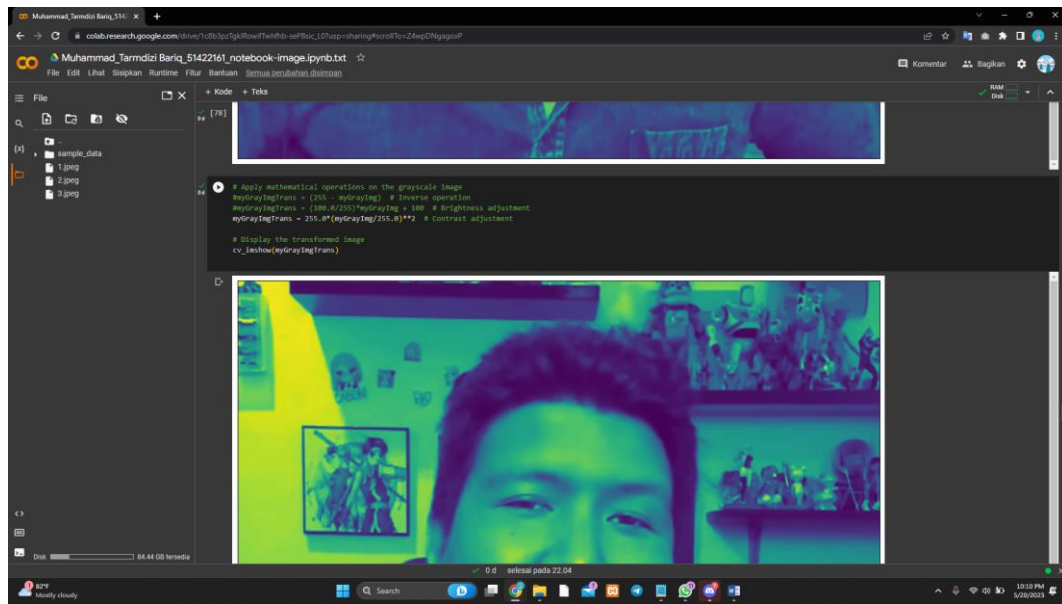
13:18 PM 5/28/2024











Muhammad\_Tamridzi\_Bariq\_514: x +

colab.research.google.com/drive/1c8b3p3gkR0wTtHfPh-s0F0uL1D?usp=sharing#scrollTo=Z4apDNagapP

Muhammad\_Tamridzi\_Bariq\_51422161\_notebook-image.ipynb.txt

File Edit Lihat Simpan Runtime Filter Bantuan (Semua perubahan disimpan)

RAM Disk

File

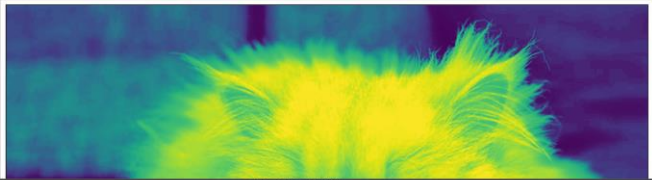
sample\_data  
1.png  
2.png  
3.png

### Histogram Equalization

Bagian ini mendemonstrasikan pemerataan histogram pada gambar gelap. Transformasi ini meratakan histogram tingkat abu-abu sehingga semua intensitas menjadi seumur mungkin. Fungsi transformasi adalah fungsi distribusi kumulatif (cdf) dari nilai piksel pada citra (dinormalisasi untuk memetakan rentang nilai piksel ke rentang yang diinginkan). Contoh ini menggunakan gambar 4 (m4).

```
# Fungsi dari histogram equalization
def histeq(im, nbr_bins = 256):
    """ Normalisasi histogram dari citra grayscale. """
    # dapatkan histogram gambar
    hist, bins = np.histogram(im.flatten(), nbr_bins, [0, 256])
    cdf = hist.cumsum() # fungsi distribusi kumulatif
    cdf = hist.max()*cdf/cdf.max() # normalisasi
    cdf_mask = np.ma.masked_equal(cdf, 0)
    cdf_mask = (cdf_mask - cdf_mask.min())*255/(cdf_mask.max()-cdf_mask.min())
    cdf = np.ma.filled(cdf_mask,0).astype('uint8')
    return cdf[im.astype('uint8')]

# terapkan fungsi pada gambar gelap Anda untuk meningkatkan kontras.
# Kita dapat mengonversi gambar ke warna untuk meningkatkan kontras.
im = histeq(im)
cv.imshow(im)
```



Muhammad\_Tamridzi\_Bariq\_514: x +

colab.research.google.com/drive/1c8b3p3gkR0wTtHfPh-s0F0uL1D?usp=sharing#scrollTo=Z4apDNagapP


Muhammad\_Tamridzi\_Bariq\_51422161\_notebook-image.ipynb.txt

File Edit Lihat Simpan Runtime Filter Bantuan (Semua perubahan disimpan)

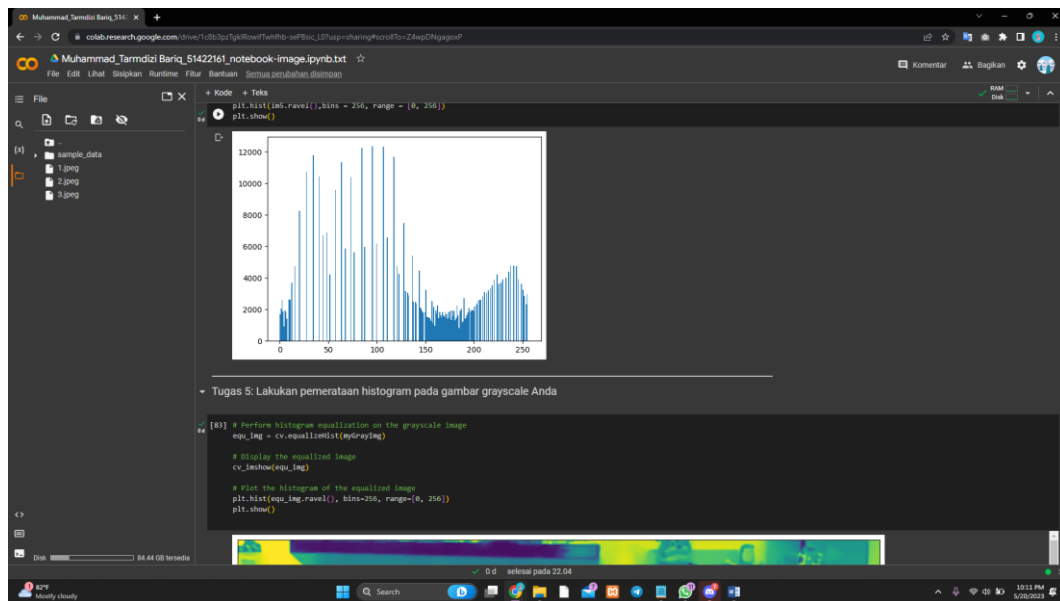
RAM Disk

File

sample\_data  
1.png  
2.png  
3.png



```
[B1] # Skema: mencoba memvisualisasikan histogram citra setelah pemerataan histogram
# Sebelum pemerataan histogram
plt.hist(im.ravel(), bins = 256, range = (0, 256))
plt.show()
```





Muhammad\_Tamridi\_Bariq\_514: K

colab.research.google.com/drive/1d83j3cTgkR0wTtHh0-s0F0wC\_LD?usp=sharing#scrollTo=Z4npDNagapsP

Muhammad\_Tamridi\_Bariq\_51422161\_notebook-image.ipynb.txt

File Edit Lihat Simpan Runtime Filter Bantuan (Semua perubahan disimpan)

File

- sample\_data
  - 1.png
  - 2.png
  - 3.png

File 0.0 GB tersedia

+ Kode + Teks

Tugas 5: Lakukan pemerataan histogram pada gambar grayscale Anda


```

# Perform histogram equalization on the grayscale image
equ_img = cv.equalizeHist(gray_img)

# Display the equalized image
cv.imshow('equ_img', equ_img)

# Plot the histogram of the equalized image
plt.hist(equ_img.ravel(), bins=256, range=[0, 256])
plt.show()

```



0.0 selesai pada 22.04

12°F Mostly cloudy

Muhammad\_Tamridi\_Bariq\_514: K

colab.research.google.com/drive/1d83j3cTgkR0wTtHh0-s0F0wC\_LD?usp=sharing#scrollTo=Z4npDNagapsP

Muhammad\_Tamridi\_Bariq\_51422161\_notebook-image.ipynb.txt

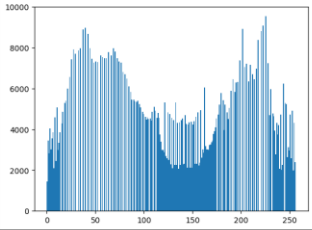
File Edit Lihat Simpan Runtime Filter Bantuan (Semua perubahan disimpan)

File

- sample\_data
  - 1.png
  - 2.png
  - 3.png

File 0.0 GB tersedia

+ Kode + Teks



5. Transformasi Fourier dari Gambar Abu-abu

Transformasi Fourier digunakan untuk mencari domain frekuensi gambar. Anda dapat menganggap gambar sebagai sinyal yang diambil sampelnya dalam dua arah. Jadi mengambil transformasi Fourier di kedua arah X dan Y memberi Anda representasi frekuensi gambar. Untuk sinyal sinusoidal, jika amplitudo berubah sangat cepat dalam waktu singkat, dapat dikatakan itu adalah sinyal frekuensi tinggi. Jika bervariasi perlahan, itu adalah sinyal frekuensi rendah. Tapi dan noise adalah konten frekuensi tinggi dalam gambar karena berubah secara drastis dalam gambar.

```

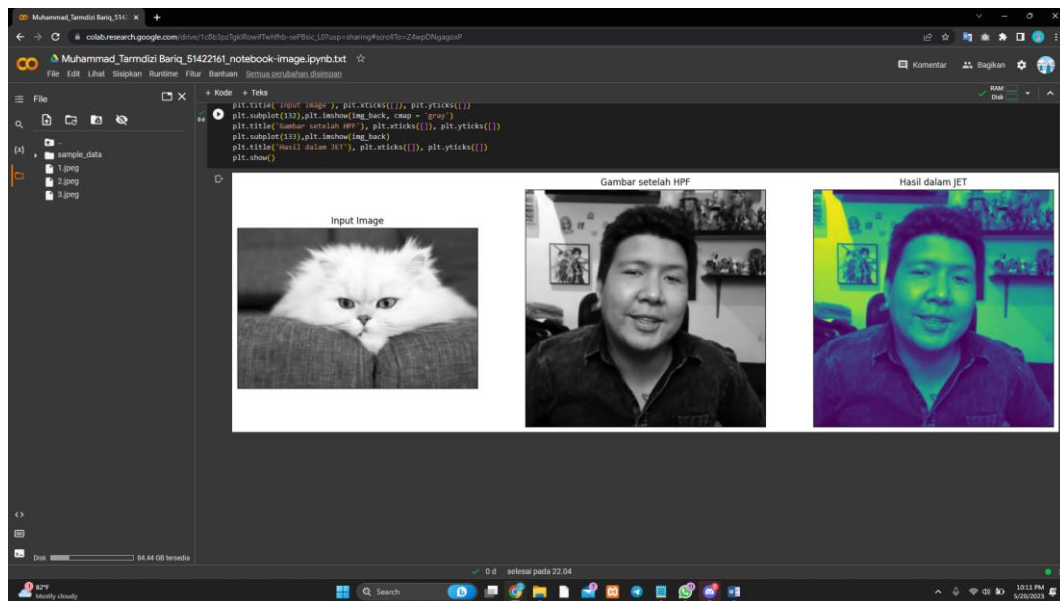
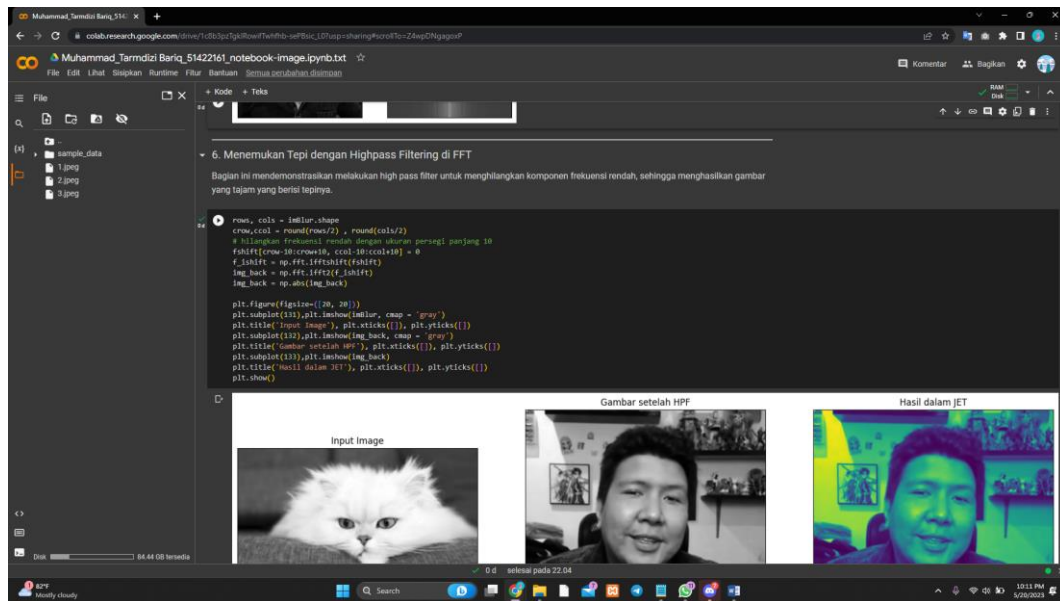
# Buramkan gambar grayscale dengan filter Gaussian dengan ukuran kernel 10
img_low = cv.bilateral_filter(img, 10, 5)
# Ubah gambar menjadi domain frekuensi
f = np.fft.fft2(img_low)
# Simetrisasi frekuensi nol ke tengah
fshift = np.fft.fftshift(f)

```

0.0 selesai pada 22.04

12°F Mostly cloudy





[https://colab.research.google.com/drive/1c8b3pzTgkIRowifTwhfhh-sePBsic\\_L0?usp=sharing#scrollTo=Z4wpDNgagoxP](https://colab.research.google.com/drive/1c8b3pzTgkIRowifTwhfhh-sePBsic_L0?usp=sharing#scrollTo=Z4wpDNgagoxP)

## 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()`.