Flip gambar

Import numpy as np # Import library numpy untuk manipulasi array

Import imageio.v2 # Import library imageio untuk membaca dan menulis gambar

Import matplotlib.pyplot as plt # Import library matplotlib untuk plotting

Img = imageio.v2.imread("GAMBAR/GTR-R35.jpg") # Membaca gambar GTR-R35.jpg ke dalam variabel img

Img\_height = img.shape[0] # Mendapatkan ukuran tinggi gambar dalam piksel

Img\_width = img.shape[1] # Mendapatkan ukuran lebar gambar dalam piksel

Img\_channel = img.shape[2] # Mendapatkan jumlah channel dalam gambar

Img\_type = img.dtype # Mendapatkan tipe data dari gambar

Img\_flip\_horizontal = np.zeros(img.shape, img\_type) # Membuat array kosong dengan ukuran yang sama dengan gambar

Img\_flip\_vertical = np.zeros(img.shape, img\_type) # Membuat array kosong dengan ukuran yang sama dengan gambar

# Melakukan flipping horizontal pada gambar

For y in range(0, img\_height):

For x in range(0, img\_width):

For c in range(0, img\_channel):

Img\_flip\_horizontal[y][x][c] = img[y][img\_width-1-x][c]

# Melakukan flipping vertical pada gambar

For y in range(0, img\_height):

For x in range(0, img\_width):

For c in range(0, img\_channel):

Img\_flip\_vertical[y][x][c] = img[img\_height-1-y][x][c]

Fig, axes = plt.subplots(1, 2, figsize=(10, 10)) # Membuat plot 2 gambar dengan ukuran 10 x 10

Ax = axes.ravel() # Melakukan plotting pada masing-masing gambar

Ax[0].imshow(img\_flip\_horizontal, cmap='gray') # Menampilkan gambar hasil flipping horizontal

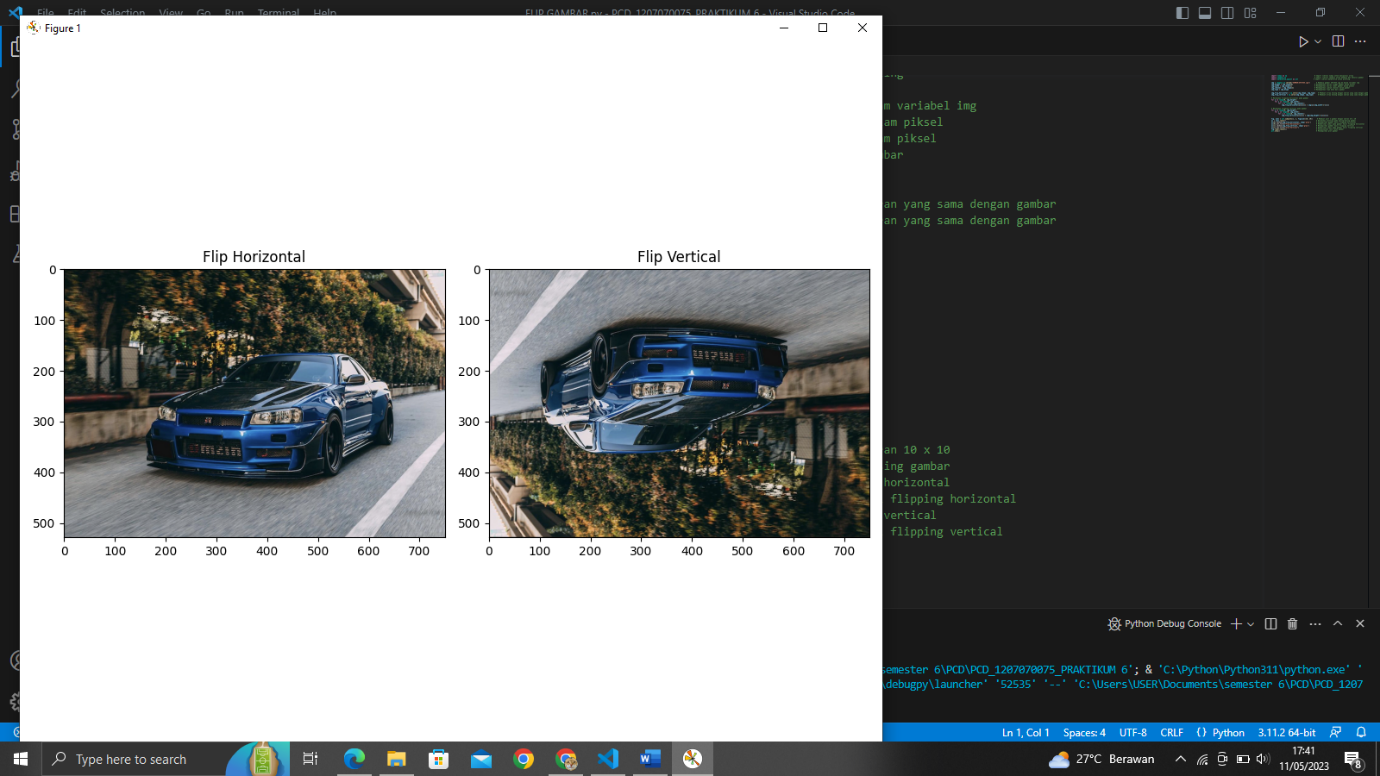
Ax[0].set\_title("Flip Horizontal") # Memberikan judul pada gambar hasil flipping horizontal

Ax[1].imshow(img\_flip\_vertical, cmap='gray') # Menampilkan gambar hasil flipping vertical

Ax[1].set\_title("Flip Vertical") # Memberikan judul pada gambar hasil flipping vertical

Fig.tight\_layout() # Menyesuaikan tata letak gambar

Plt.show() # Menampilkan plot gambar



Histogram

Import numpy as np # Import library numpy untuk manipulasi array

Import imageio # Import library imageio untuk membaca dan menulis gambar

Import matplotlib.pyplot as plt # Import library matplotlib untuk plotting

Img = imageio.imread("GAMBAR/GTR-R35.jpg")

Img\_height = img.shape[0]

Img\_width = img.shape[1]

Img\_channel = img.shape[2]

Img\_grayscale = np.zeros(img.shape, dtype=np.uint8)

For y in range(0, img\_height):

For x in range(0, img\_width):

Red = img[y][x][0]

Green = img[y][x][1]

Blue = img[y][x][2]

Gray = (int(red) + int(green) + int(blue)) / 3

Img\_grayscale[y][x] = (gray, gray, gray)

Plt.imshow(img\_grayscale)

Plt.title("Grayscale")

Plt.show()

Hg = np.zeros((256)) #Membuat variabel untuk menyimpan data gambar

For x in range(0, 256):

Hg[x] = 0

For y in range(0, img\_height):

For x in range(0, img\_width):

Gray = img\_grayscale[y][x][0]

Hg[gray] += 1

Plt.figure(figsize=(20, 6))

Plt.plot(hg, color="black", linewidth=2.0)

Plt.show()

Bins = np.linspace(0, 256, 100)

Plt.hist(hg, bins, color="black", alpha=0.5)

Plt.title("Histogram")

Plt.show()

For x in range(0, 256):

Hgr[x] = 0

Hgg[x] = 0

Hgb[x] = 0

For x in range(0, 768):

Hgrgb[x] = 0

For x in range(0, 256):

Hgr[x] = 0 # Set nilai awal histogram untuk channel merah

Hgg[x] = 0 # Set nilai awal histogram untuk channel hijau

Hgb[x] = 0 # Set nilai awal histogram untuk channel biru

For x in range(0, 768):

Hgrgb[x] = 0 # Set nilai awal histogram untuk keseluruhan gambar

Th = int(256/64) # Tentukan threshold untuk histogram

Temp = [0] # Buat array kosong untuk menyimpan nilai sementara

For y in range(0, img.shape[0]):

For x in range(0, img.shape[1]):

Red = int(img[y][x][0]) # Ambil nilai untuk channel merah

Green = int(img[y][x][1]) # Ambil nilai untuk channel hijau

Blue = int(img[y][x][2]) # Ambil nilai untuk channel biru

Green = green + 256 # Shift nilai hijau sebesar 256

Blue = blue + 512 # Shift nilai biru sebesar 512

Temp.append(green) # Tambahkan nilai hijau ke dalam array temp

Hgrgb[red] += 1 # Tambahkan nilai untuk channel merah pada histogram

Hgrgb[green] += 1 # Tambahkan nilai untuk channel hijau pada histogram

Hgrgb[blue] += 1 # Tambahkan nilai untuk channel biru pada histogram

Binsrgb = np.linspace(0, 768, 100) # Tentukan jarak antar bin pada histogram RGB

Plt.hist(hgrgb, binsrgb, color="black", alpha=0.5) # Plot histogram

Plt.plot(hgrgb) # Tampilkan plot histogram

Plt.title("Histogram Red Green Blue") # Set judul pada plot

Plt.show() # Tampilkan plot

For y in range(0, img\_height):

For x in range(0, img\_width):

Red = img[y][x][0]

Green = img[y][x][1]

Blue = img[y][x][2]

Hgr[red] += 1

Hgg[green] += 1

Hgb[blue] += 1

Bins = np.linspace(0, 256, 100)

Plt.hist(hgr, bins, color="red", alpha=0.5)

Plt.title("Histogram Red")

Plt.show()

Plt.hist(hgg, bins, color="green", alpha=0.5)

Plt.title("Histogram Green")

Plt.show()

Plt.hist(hgb, bins, color="blue", alpha=0.5)

Plt.title("Histogram Blue")

Plt.show()

Hgk = np.zeros((256))

C = np.zeros((256))

For x in range(0, 256):

    Hgk[x] = 0

    C[x] = 0

For y in range(0, img\_height):

    For x in range(0, img\_width):

        Gray = img\_grayscale[y][x][0]

        Hgk[gray] += 1

C[0] = hgk[0]

For x in range(1, 256):

     C[x] = c[x-1] + hgk[x]

Hmaxk = c[255]

For x in range(0, 256):

    C[x] = 190 \* c[x] / hmaxk

Plt.hist(c, bins, color="black", alpha=0.5)

Plt.title("Histogram Grayscale Kumulatif")

Plt.show()

Hgh = np.zeros((256))

H = np.zeros((256))

C = np.zeros((256))

For x in range(0, 256):

    Hgh[x] = 0

    H[x] = 0

    C[x] = 0

For y in range(0, img\_height):

    For x in range(0, img\_width):

        Gray = img\_grayscale[y][x][0]

        Hgh[gray] += 1

H[0] = hgh[0]

For x in range(1, 256):

     H[x] = h[x-1] + hgh[x]

For x in range(0, 256):

     H[x] = h[x] / img\_height / img\_width

For x in range(0, 256):

    Hgh[x] = 0

For y in range(0, img\_height):

    For x in range(0, img\_width):

        Gray = img\_grayscale[y][x][0]

        Gray = h[gray] \* 255

        Hgh[int(gray)] += 1

C[0] = hgh[0]

For x in range(1, 256):

     C[x] = c[x-1] + hgh[x]

Hmaxk = c[255]

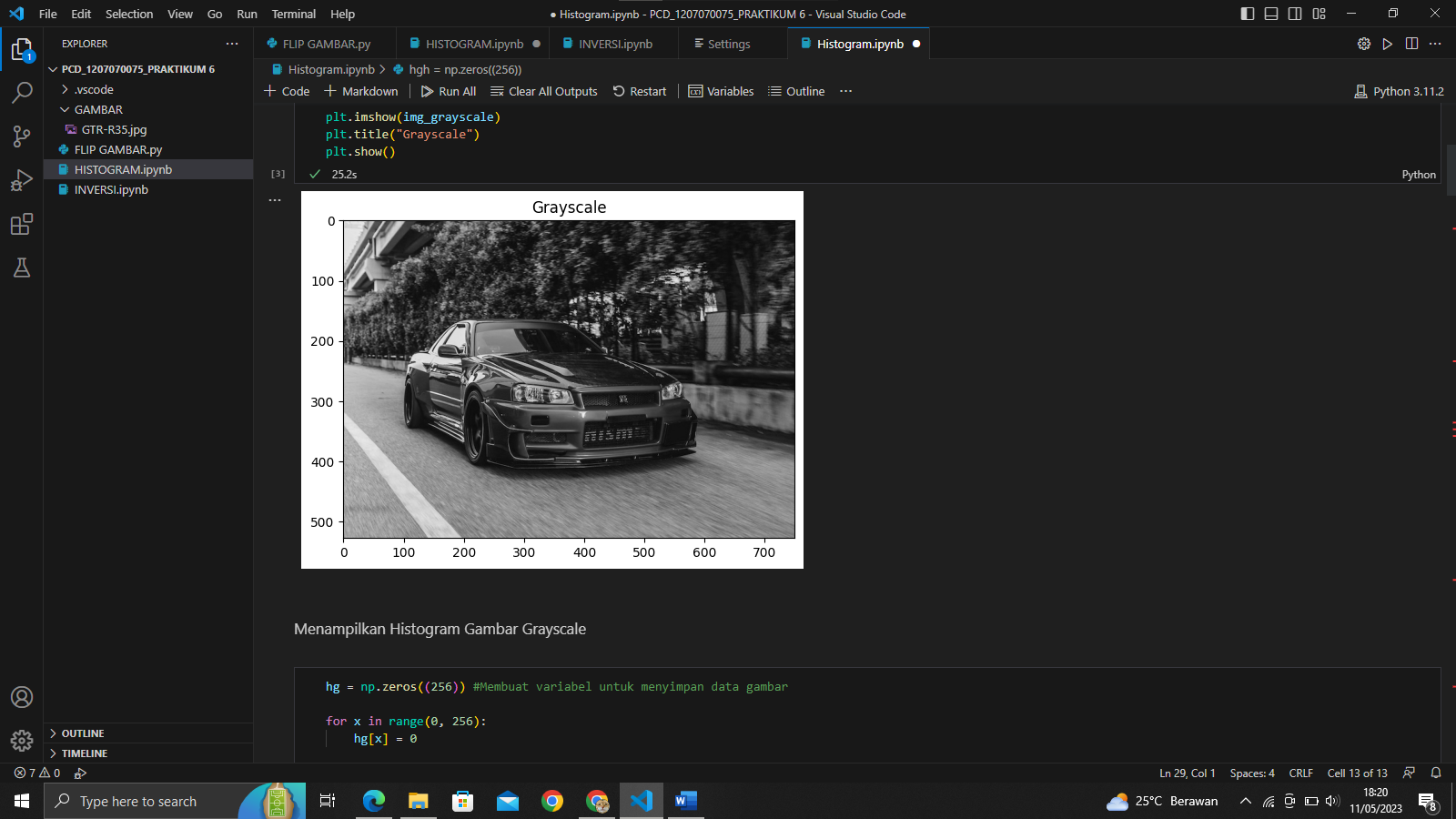
For x in range(0, 256):

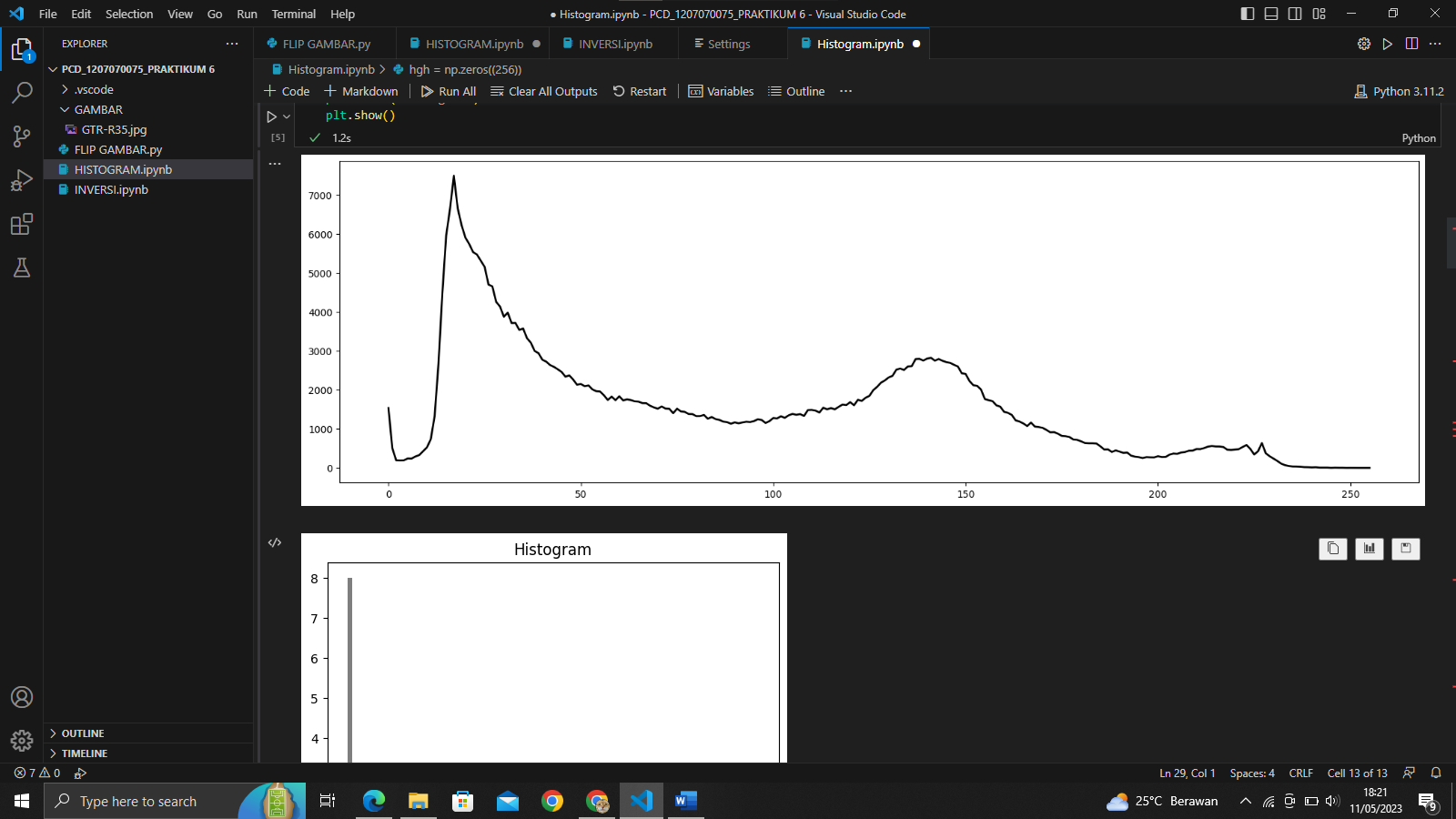
    C[x] = 190 \* c[x] / hmaxk

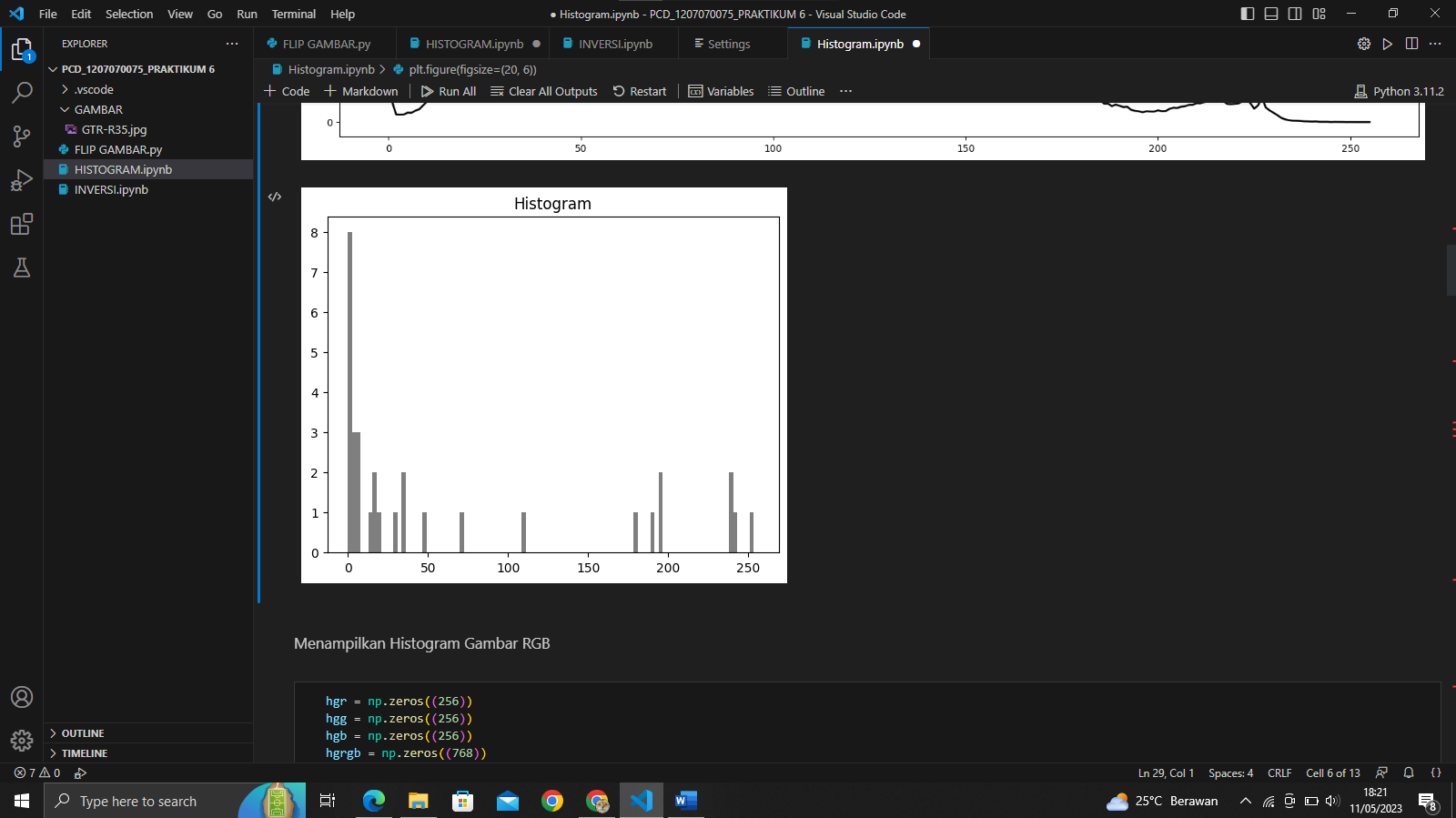
Plt.hist(c, bins, color="black", alpha=0.5)

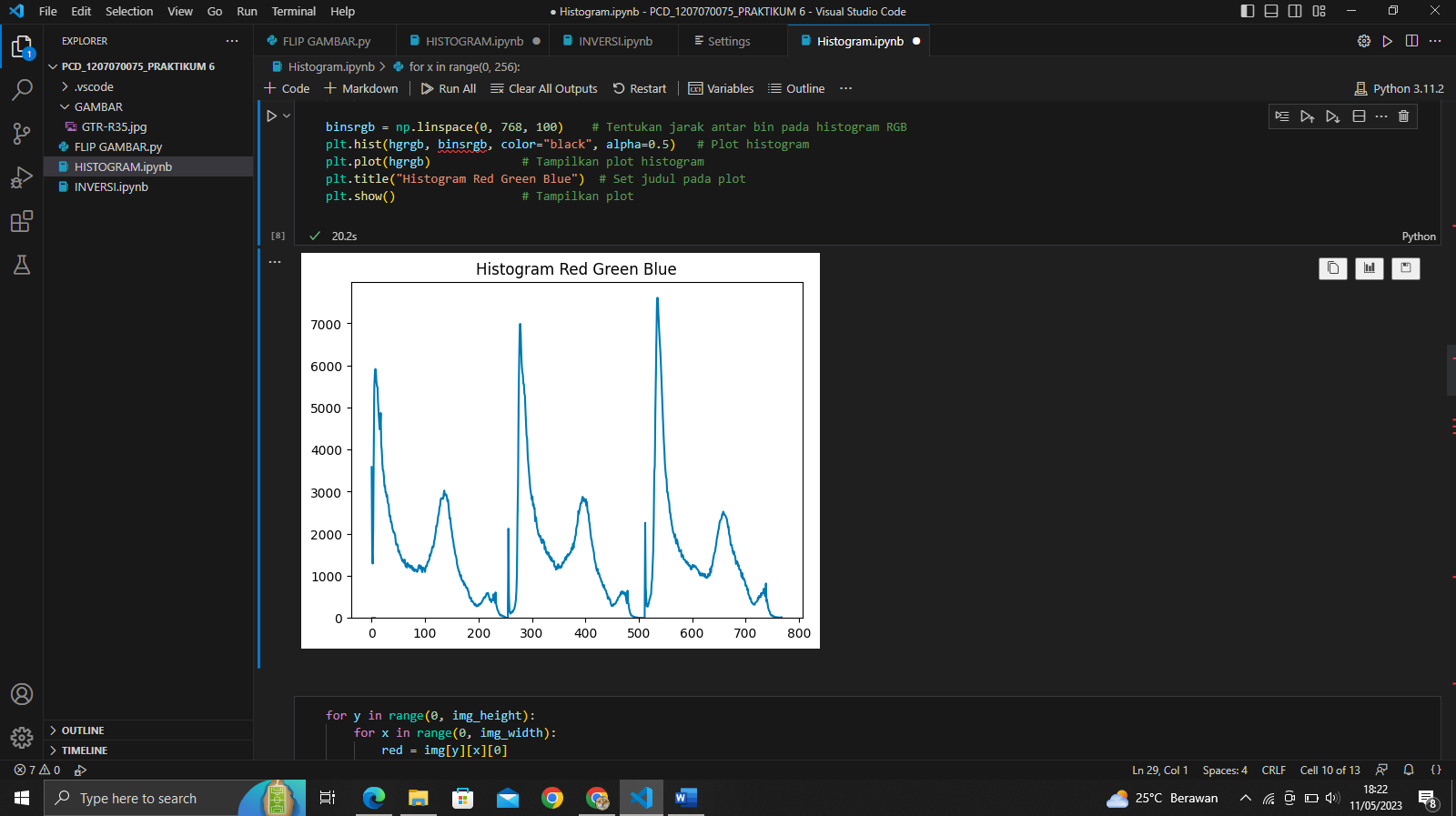
Plt.title("Histogram Grayscale Hequalisasi")

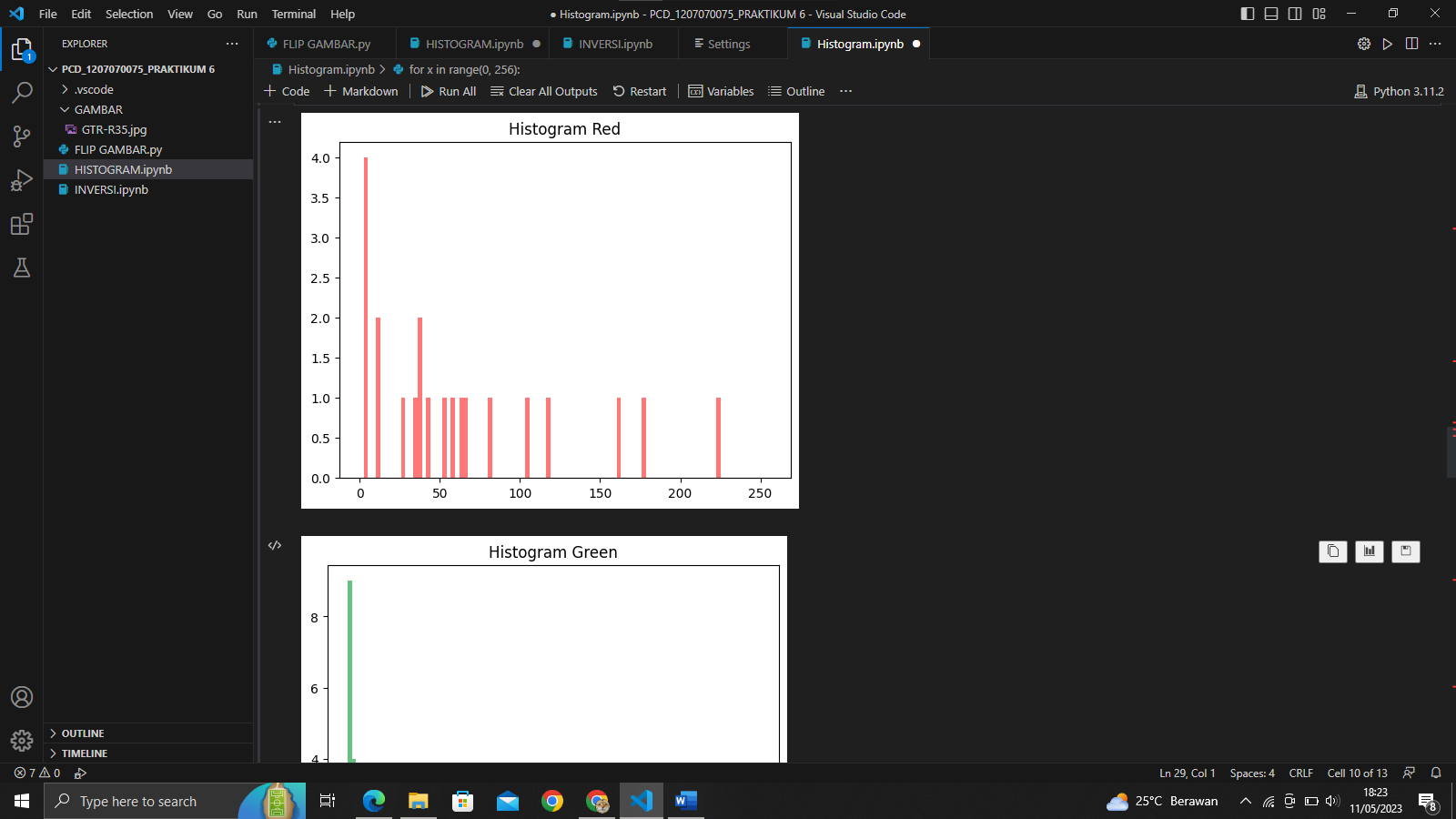
Plt.show()

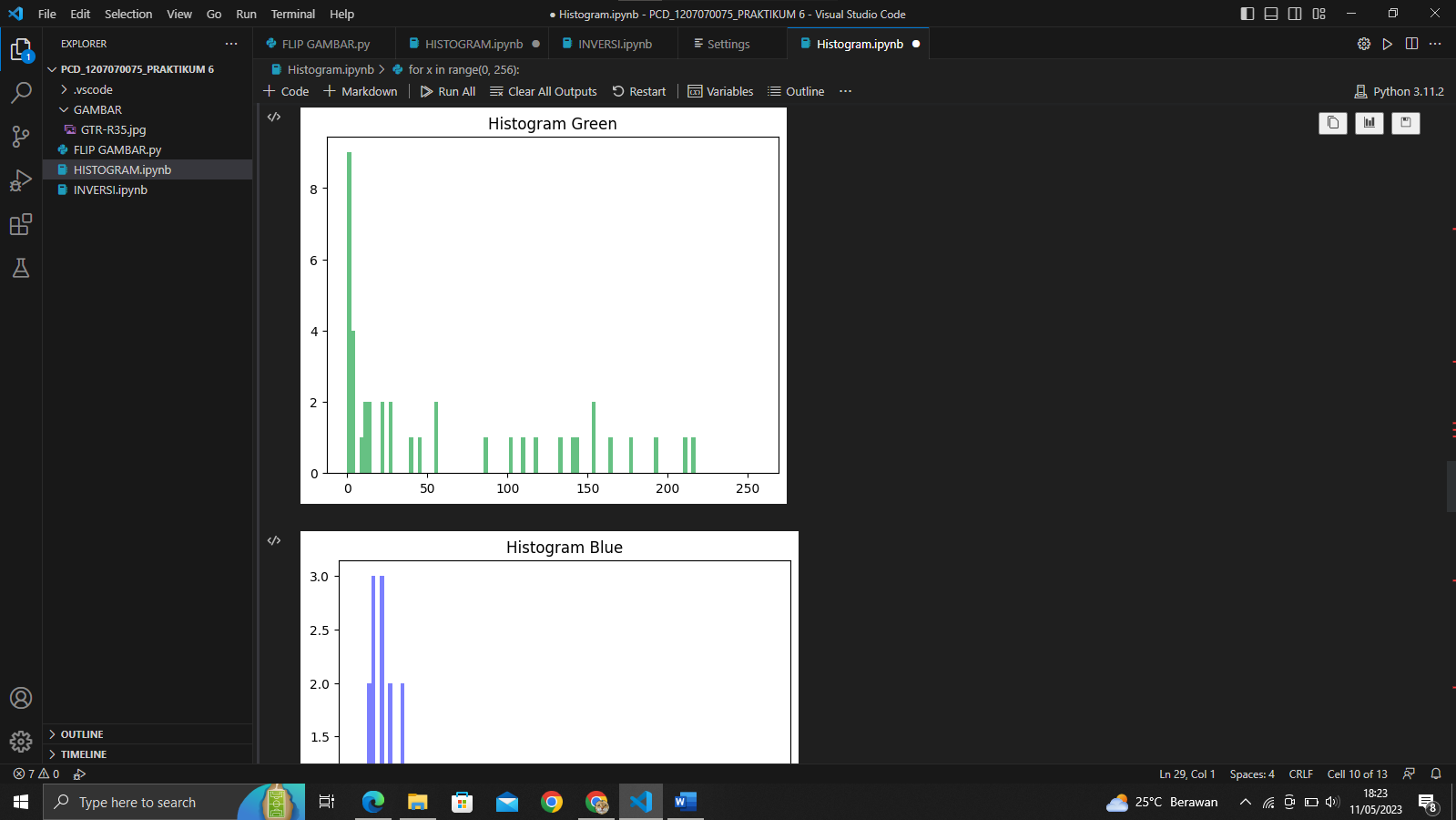


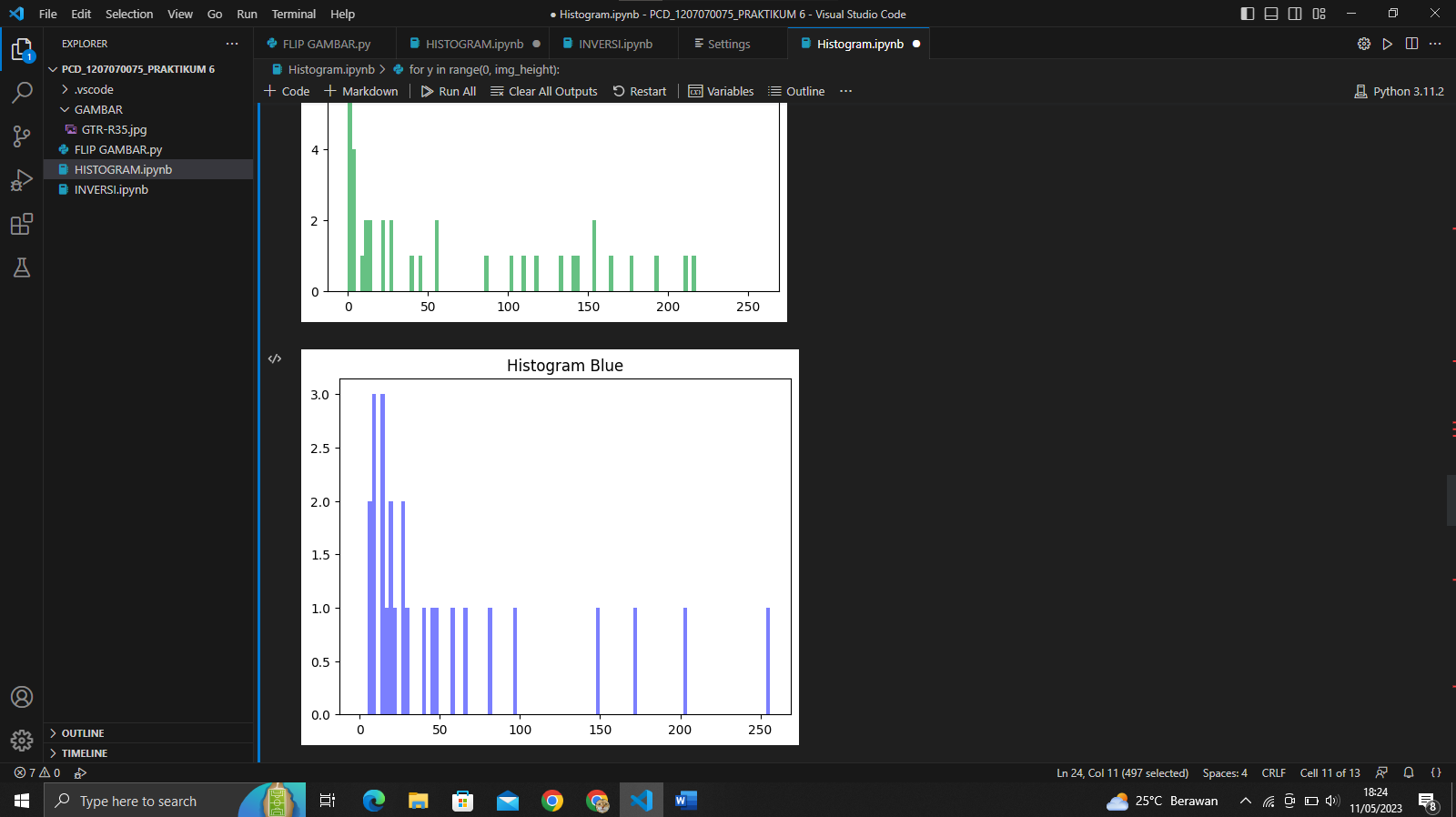


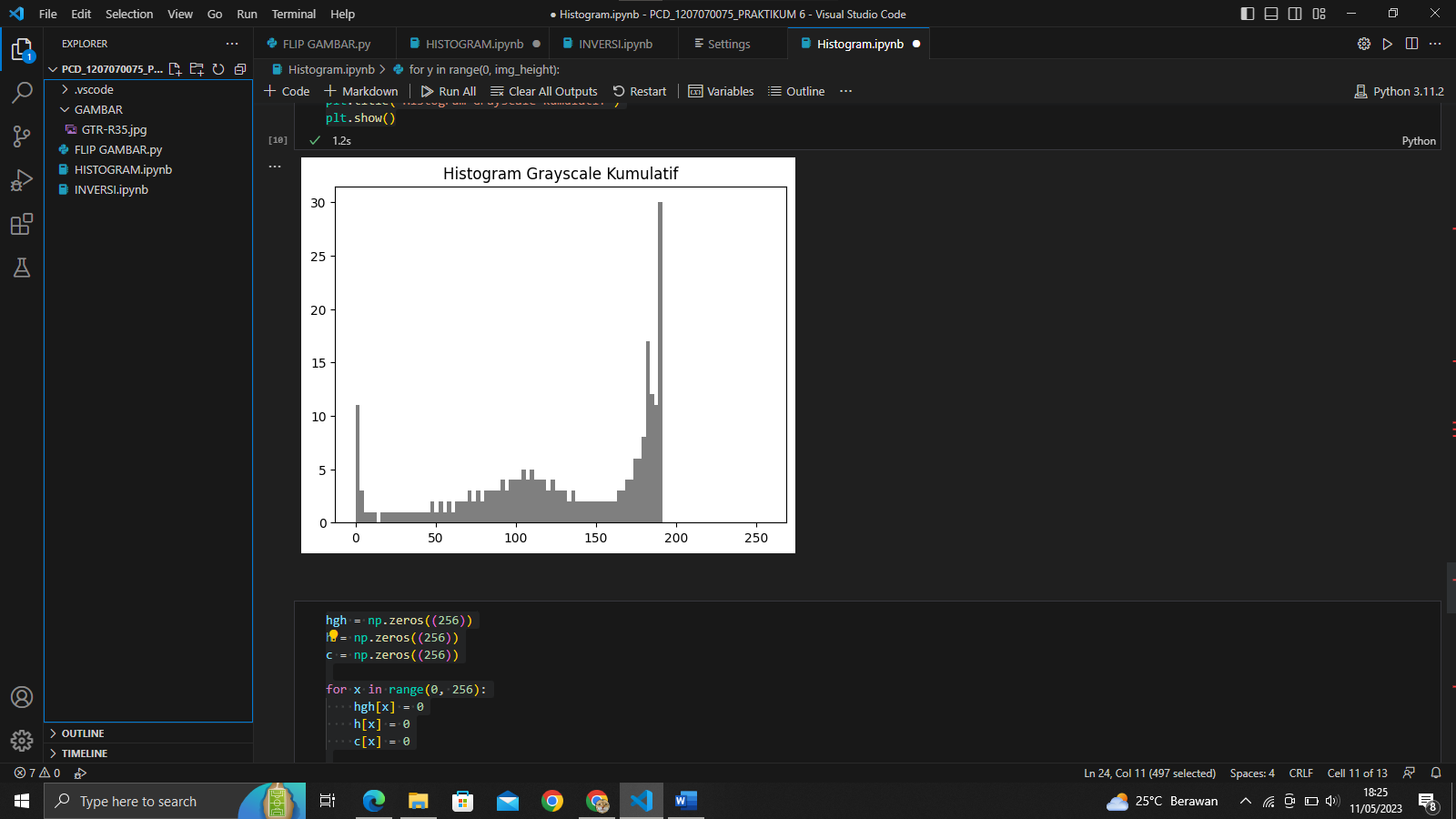


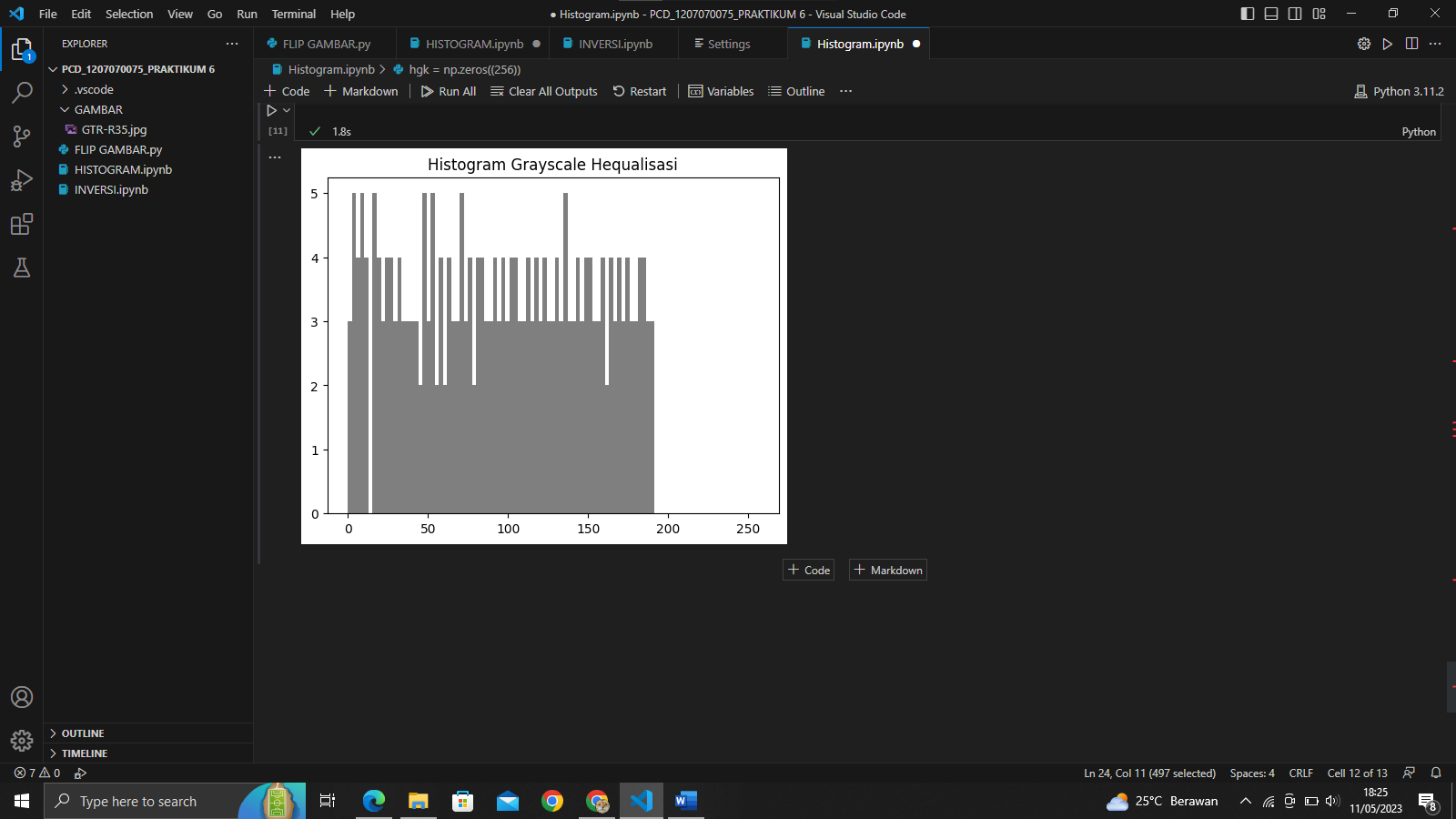












Inversi

import numpy as np #melakukan pemanggilan library numpy dengan alias np

import cv2 #untuk menampilkan membaca dan menulis gambar

import matplotlib.pyplot as plt  #melakukan pemanggilan library untuk melakukan visualisasi dan menampilkan gambar atau hasil

img = cv2.imread("GAMBAR/GTR-R35.jpg") #melakukan pembacaan gambar yang berada satu folder

img\_height = img.shape[0] #untuk mendapatkan ukuran lebar dari gambar

img\_width = img.shape[1] #untuk mendapatkan ukuran tinggi dari gambar

img\_channel = img.shape[2] #untuk mendapatkan resolusi gambar

img\_inversi = np.zeros(img.shape, dtype=np.uint8) #membuat variable untuk inversi sesuai type data

def inversi\_grayscale(nilai): #untuk menyatakan fungsi inversi grayscale

    for y in range(0, img\_height): #memberikan interval range untuk variabel y sebagai ketinggian

        for x in range(0, img\_width): #memberikan interval range untuk variabel x sebagai lebar gambar

            red = img[y][x][0] #memberikan nilai piel untuk red

            green = img[y][x][1] #memberikan nilai pixel untuk green

            blue = img[y][x][2] #memberikan nilai pixel untuk blue

            gray = (int(red) + int(green) + int(blue)) / 3 #rumus untuk merubah gambar original menjadi grayscale

            gray = nilai - gray #rumus untuk menentukan hasil yang ditampilkan

            img\_inversi[y][x] = (gray, gray, gray) #mnampilkan hasil dari prubahan

def inversi\_rgb(nilai):#untuk menyatakan fungsi inversi rgb

    for y in range(0, img\_height): #memberikan interval range untuk variabel y sebagai ketinggian

        for x in range(0, img\_width):#memberikan interval range untuk variabel x sebagai lebar gambar

            red = img[y][x][0]#menentukan variable untuk red x,y

            red = nilai - red #menentukan hasil red yang dikurangi nilai untuk inversi

            green = img[y][x][1]#menentukan variable untuk green x,y

            green = nilai - green#menentukan hasil green yang dikurangi nilai untuk inversi

            blue = img[y][x][2]#menentukan variable untuk blue x,y

            blue = nilai - blue#menentukan hasil bllue yang dikurangi nilai untuk inversi

            img\_inversi[y][x] = (red, green, blue)#menampilkan hasil output dari perubahan

inversi\_grayscale(255) #menampilkan nilai pixel inversi grayscale

plt.imshow(img\_inversi) #menampilkan gambar hasil inversi

plt.title("Inversi Grayscale") #memasukkan judul untuk gambar

plt.show()#mnampilkan keseluruhan gambar yang akan ditampilkan di window

inversi\_rgb(255) #menampilkan nilai pixel rgb hasil inversi

plt.imshow(img\_inversi)#menamopilkan gambar hasil dari inversi rgb

plt.title("Inversi RGB") #mmasukkan judul gambar yang akan ditampilkan

plt.show()#menampilkan keseluruhan gambar yang akan ditampilkan di windows

#inversi & log

def inlog(c): #mendifinisikan fungsi untuk inversi inlog

    for y in range(0, img\_height):#memberikan nilai interval range untuk variabel y sebagai ketinggian gambar

        for x in range(0, img\_width):#memberikan nilai interrval range untuk variable x sebagai lebar gambar

            red = img[y][x][0] #memberikan nilai piel untuk red

            green = img[y][x][1] #memberikan nilai pixel untuk green

            blue = img[y][x][2] #memberikan nilai pixel untuk blue

            gray = (int(red) + int(green) + int(blue)) / 3 #rumus untuk merubah gambar original menjadi grayscale

            gray = int(c \* np.log(255 - gray + 1))#rumus untuk invers log

            if gray > 255: #penentuan gray menggunakan fungsi if

                gray = 255 #nilai untuk gray

            if gray < 0:#penentuan nilai gray  menggunakan fungsi if

                gray = 0#nilai untuk gray

            img\_inlog[y][x] = (gray, gray, gray) #hasil yang keluar sebagai output

inlog(30) #nilai inlog yang di tampilkan

plt.imshow(img\_inlog)#penampilan hasil gambar inversi log

plt.title("Inversi & Log")#memasukkan atau memberikan judul pada gambar yang akan ditampilkan

plt.show()#menampilkan keseluruhan gambar pada windows

img\_nthpower = np.zeros(img.shape, dtype=np.uint8) #membuat variable img\_nthPower untuk menampung hasil

def nthpower(c, y): #mendefinisikan fungsi x,y untuk nth power

    thc = c / 100 #penentuan nilai thc

    thy = y / 100 #penentuan nilai thy

    for y in range(0, img\_height): #memberikan interval range untuk variable y sebagai ketinggian gambar

        for x in range(0, img\_width): #memberikan interval range  untuk variable y sebagai lebar gambar

            red = img[y][x][0] #memberikan nilai piel untuk red

            green = img[y][x][1] #memberikan nilai pixel untuk green

            blue = img[y][x][2] #memberikan nilai pixel untuk blue

            gray = (int(red) + int(green) + int(blue)) / 3 #rumus untuk merubah gambar original menjadi grayscale

            gray = int(thc \* pow(gray, thy)) #rumus untuk nth powr

            if gray > 255: #penentuan nilai untuk gray menggunakan fungsi if

                gray = 255 #nilai gray

            if gray < 0: #penentuan nilai untuk gray menggunakan fungsi if

                gray = 0 #nilai gray

            img\_nthpower[y][x] = (gray, gray, gray) #hasil gambar yang akan ditampilkan

nthpower(50, 100) #nilai nth yang akan ditampilkan

plt.imshow(img\_nthpower) #penampilan gambar hasil nth power

plt.title("Nth Power")#untuk memasukkan judul pada gambar yang akan di tampilkan

plt.show() #menampilkan keseluruhan gambar hasil nth power yang ditampilkan di window

#nth root power

img\_nthrootpower = np.zeros(img.shape, dtype=np.uint8) #membuat variable img\_nthrootpower

def nthrootpower(c, y): #mendefinisikan fungsi untuk nth root power

    thc = c / 100 #penentuan nilai thc

    thy = y / 100 #pnentuan nilai thy

    for y in range(0, img\_height): #memberikan interval range untuk variable y sebagai ketinggian  gambar

        for x in range(0, img\_width): #memberikan interval range untuk variasi x sebagai lebar gambar

            red = img[y][x][0] #memberikan nilai piel untuk red

            green = img[y][x][1] #memberikan nilai pixel untuk green

            blue = img[y][x][2] #memberikan nilai pixel untuk blue

            gray = (int(red) + int(green) + int(blue)) / 3 #rumus untuk merubah gambar original menjadi grayscale

            gray = int(thc \* pow(gray, 1./thy)) #rumus untuk nth root power

            if gray > 255:#menentukan nilai gray menggunakan fungsi if

                gray = 255 #nilai gray yang harus ditampilkan

            if gray < 0:#menentukan nilai gray menggunakan fungsi ig

                gray = 0 #nilai yang harus ditampilkan

            img\_nthpower[y][x] = (gray, gray, gray) #hasil gambar yang akan ditampilkan

nthrootpower(50, 100) #nilai nth root power

plt.imshow(img\_nthrootpower)#menampilkan gambar hasil nthrootpower

plt.title("Nth Root Power")#memasukkan judul pada gambar yang akan dita,pilkan

plt.show()#menampilkan keseluruhan gambar hasil yang akan ditampilkan di windows

