Nama : Muhamad Arif Maulana

Nim : 1207070069

PCD TSEB 20

**Flip Gambar**

Source Code

import numpy as np #memanggil library numpy

import imageio #memanggil library imageio

import matplotlib.pyplot as plt #memanggil library matplotlib

img = imageio.imread("foto/R25.jpg") #memanggil gambar yang terdapat dalam folder

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

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

img\_channel = img.shape[2] #untuk mengisi warna pada gambar

img\_type = img.dtype #untuk membaca gambar

img\_flip\_horizontal = np.zeros(img.shape, img\_type) #untuk menghorizontalkan tampilan gambar

img\_flip\_vertical = np.zeros(img.shape, img\_type) #untuk memverticalkan tampilan gambar

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

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

        for c in range(0, img\_channel): #untuk memberikan interval range pada c sebagai warna

            img\_flip\_horizontal[y][x][c] = img[y][img\_width-1-x][c] #untuk memberikan eksekusi horizontal gambar dari lebar gambar

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

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

        for c in range(0, img\_channel): #untuk memberikan interval range pada c sebagai warna

            img\_flip\_vertical[y][x][c] = img[img\_height-1-y][x][c] #untuk memberikan eksekusi vertical gambar dari tinggi gambar

plt.imshow(img\_flip\_horizontal) #untuk memanggil gambar yang sudah di horizontalkan

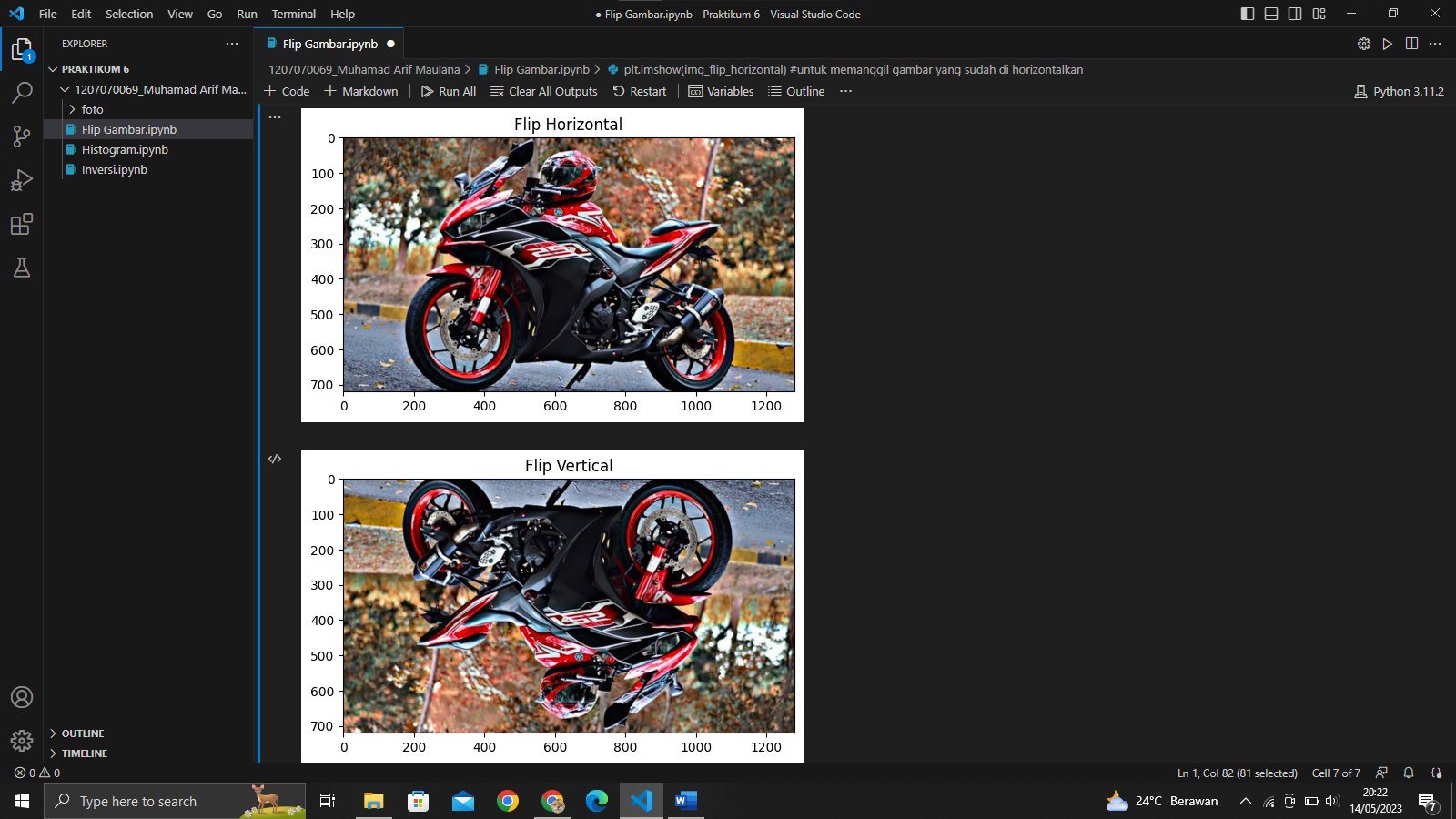
plt.title("Flip Horizontal") #untuk memberi nama gambar

plt.show() #untuk menampilkan gambar

plt.imshow(img\_flip\_vertical) #untuk memanggil gambar yang sudah di verticalkan

plt.title("Flip Vertical") #untuk memberi nama gambar

plt.show() #untuk menampilkan gambar



**Histogram**

Source code

import numpy as np #memanggil library numpy

import imageio #memanggil library image io

import matplotlib.pyplot as plt #memanggil library matplotlib

img = imageio.imread("foto/arif.jpg") #memanggil gambar pada folder

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

img\_width = img.shape[1] #untuk mendapatkan lebar gambar

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

img\_grayscale = np.zeros(img.shape, dtype=np.uint8) #mengosongkan aray untuk menyimpan pixel

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

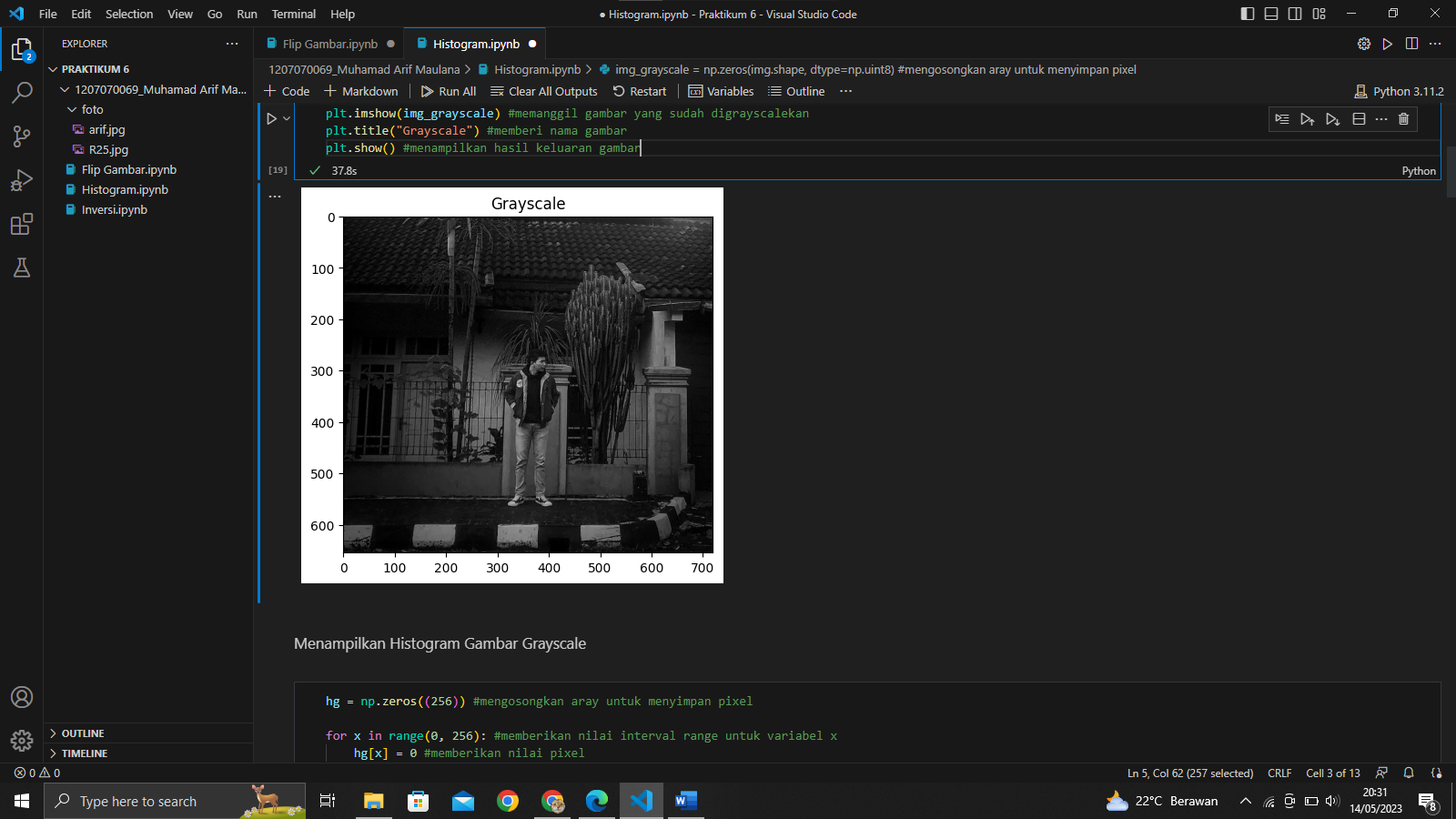
        gray = (int(red) + int(green) + int(blue)) / 3 #rumus untuk merubah gambar original menjadi grayscale

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

plt.imshow(img\_grayscale) #memanggil gambar yang sudah digrayscalekan

plt.title("Grayscale") #memberi nama gambar

plt.show() #menampilkan hasil keluaran gambar



hg = np.zeros((256)) #mengosongkan aray untuk menyimpan pixel

for x in range(0, 256): #memberikan nilai interval range untuk variabel x

    hg[x] = 0 #memberikan nilai pixel

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

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

        gray = img\_grayscale[y][x][0] #memberikan nilai pixel untuk gray

        hg[gray] += 1 #

plt.figure(figsize=(20, 6)) #mengatur ukuran hasil gambar keluaran

plt.plot(hg, color="black", linewidth=2.0) #untuk menampilkan warna hitam pada histogram

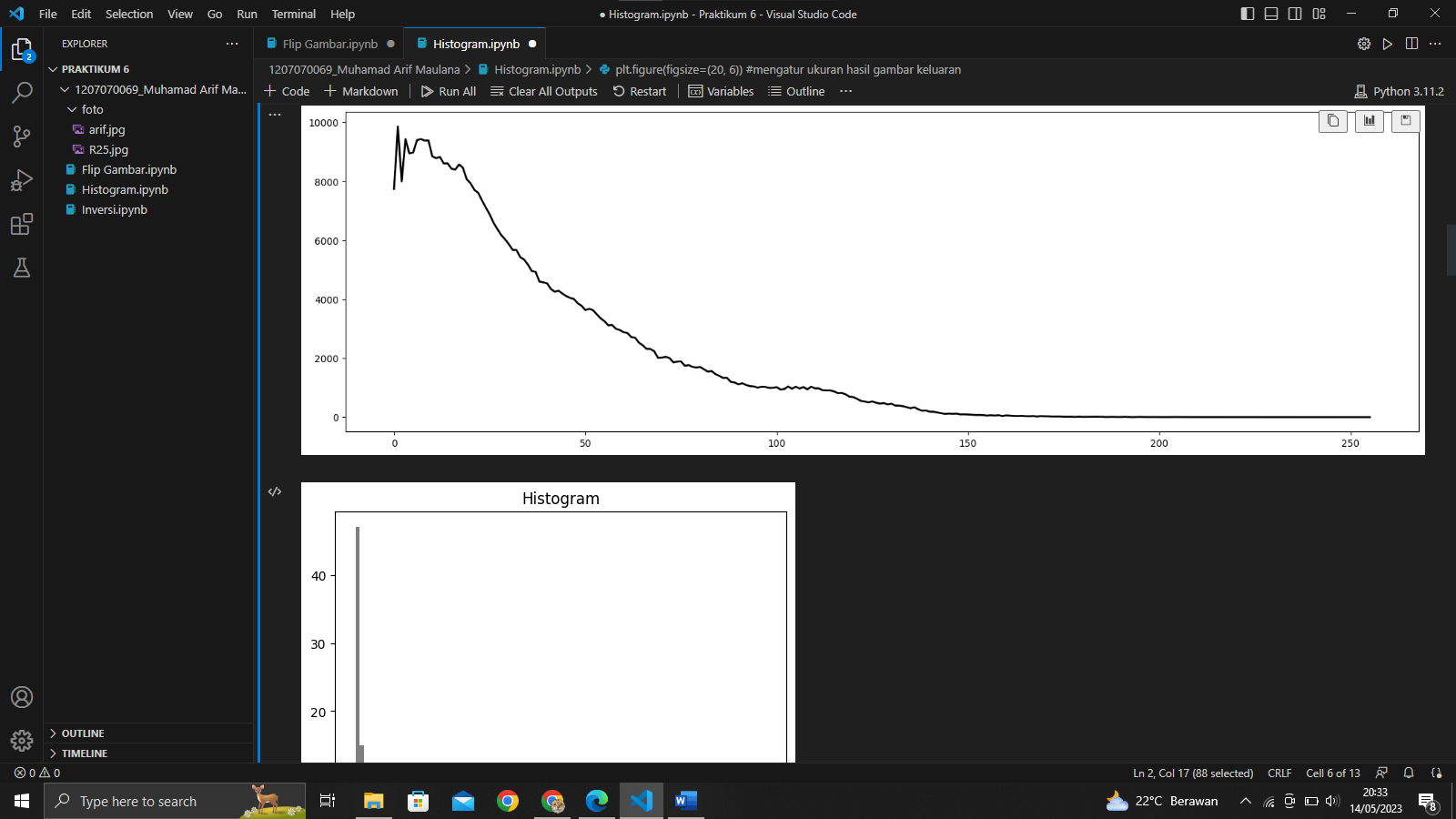
plt.show() #menampilkan keluaran hasil gambar

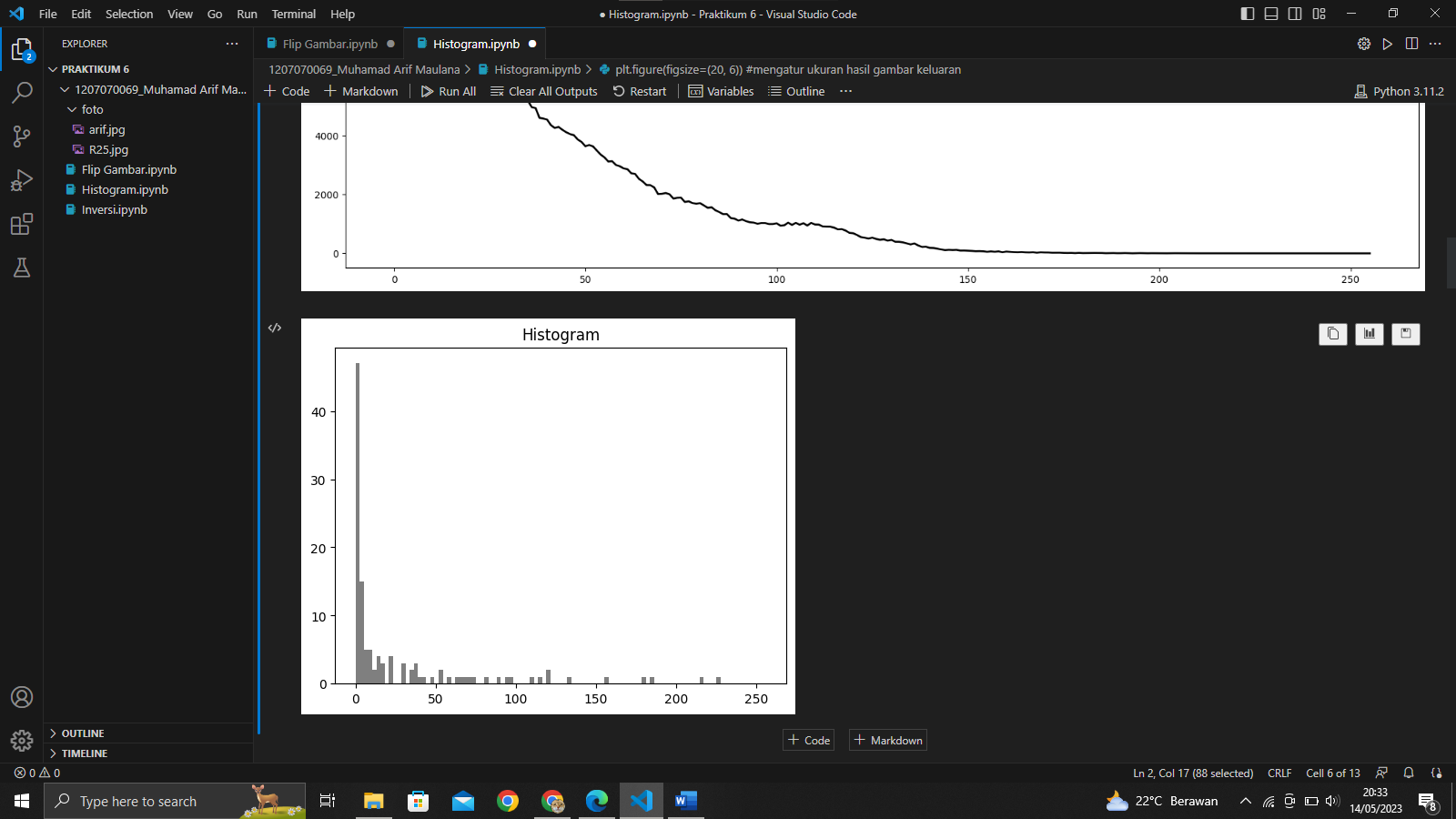
bins = np.linspace(0, 256, 100)

plt.hist(hg, bins, color="black", alpha=0.5) #untuk menampilkan histogram warna hitam saja pada histogram

plt.title("Histogram") #memberi nama keluaran hasil gambar

plt.show() #menampilkan keluaran hasil gambar





for x in range(0, 256): #memberikan nilai interval range untuk variabel x

    hgr[x] = 0 #untuk menghapus kemunculan pixel red dari histogram dengan mengatur nilai menjadi 0

    hgg[x] = 0 #untuk menghapus kemunculan pixel green dari histogram dengan mengatur nilai menjadi 0

    hgb[x] = 0 #untuk menghapus kemunculan pixel blue dari histogram dengan mengatur nilai menjadi 0

for x in range(0, 768): #memberikan nilai interval range untuk variabel x

    hgrgb[x] = 0 #untuk menghapus kemunculan pixel rgb dari histogram dengan mengatur nilai menjadi 0

for x in range(0, 256): #memberikan nilai interval range untuk variabel x

    hgr[x] = 0 #untuk menghapus kemunculan pixel red dari histogram dengan mengatur nilai menjadi 0

    hgg[x] = 0 #untuk menghapus kemunculan pixel green dari histogram dengan mengatur nilai menjadi 0

    hgb[x] = 0 #untuk menghapus kemunculan pixel blue dari histogram dengan mengatur nilai menjadi 0

for x in range(0, 768): #memberikan nilai interval range untuk variabel x

    hgrgb[x] = 0 #untuk menghapus kemunculan pixel rgb dari histogram dengan mengatur nilai menjadi 0

th = int(256/64)

temp = [0]

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 pixel untuk red

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

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

        green = green + 256 #menambahkan 256 untuk green

        blue = blue + 512 #menambahkan 512 untuk green

        temp.append(green)

        hgrgb[red] += 1

        hgrgb[green] += 1

        hgrgb[blue] += 1

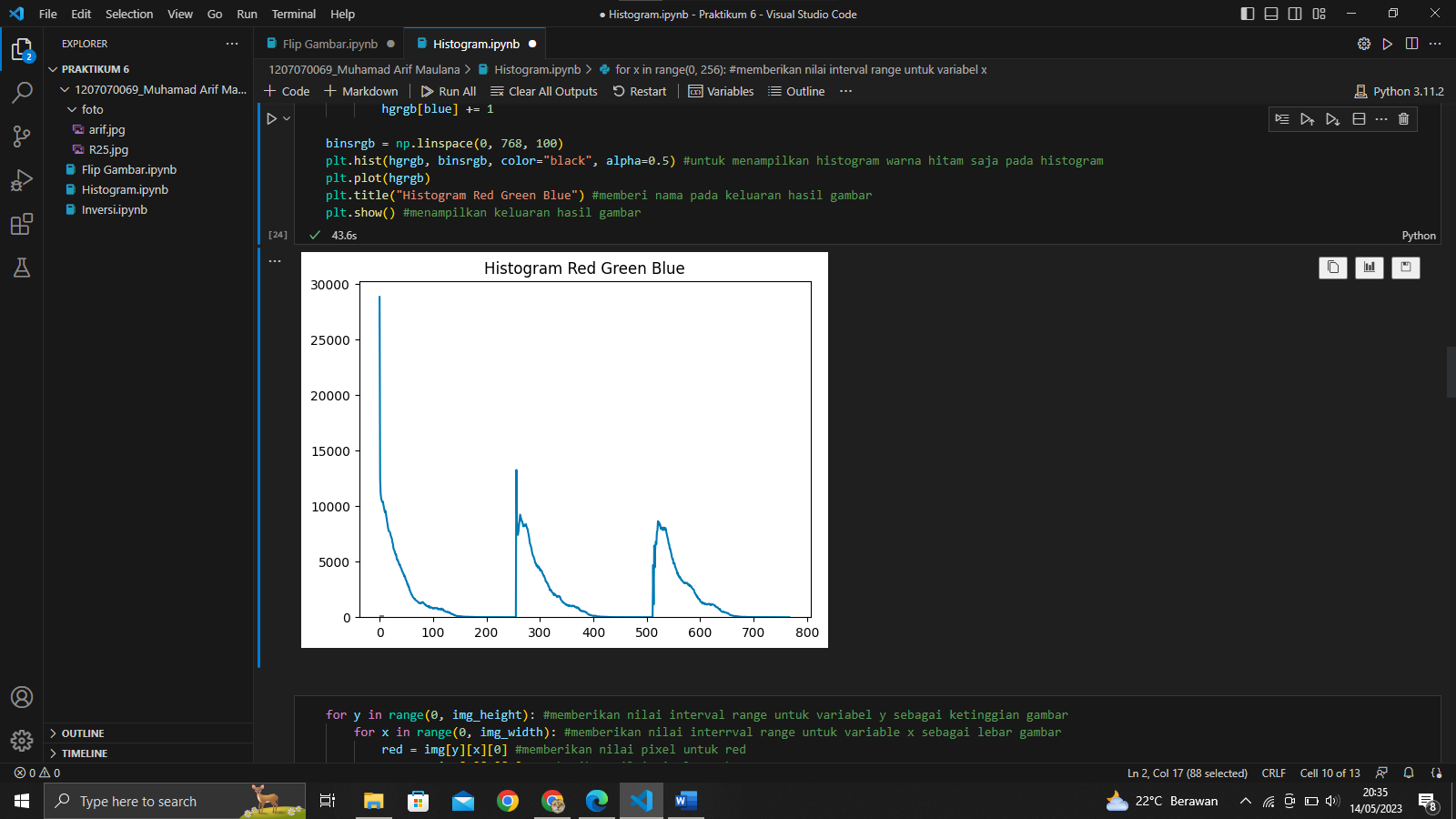
binsrgb = np.linspace(0, 768, 100)

plt.hist(hgrgb, binsrgb, color="black", alpha=0.5) #untuk menampilkan histogram warna hitam saja pada histogram

plt.plot(hgrgb)

plt.title("Histogram Red Green Blue") #memberi nama pada keluaran hasil gambar

plt.show() #menampilkan keluaran hasil gambar



for y in range(0, img\_height): #memberikan nilai interval range untuk variable 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 pixel untuk red

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

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

        hgr[red] += 1

        hgg[green] += 1

        hgb[blue] += 1

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

plt.hist(hgr, bins, color="red", alpha=0.5) #menampilkan histogram dengan warna merah saja pada histogram

plt.title("Histogram Red") #memberi nama pada gambar

plt.show() #menampilkan keluaran hasil gambar

plt.hist(hgg, bins, color="green", alpha=0.5) #menampilkan histogram dengan warna hijau saja pada histogram

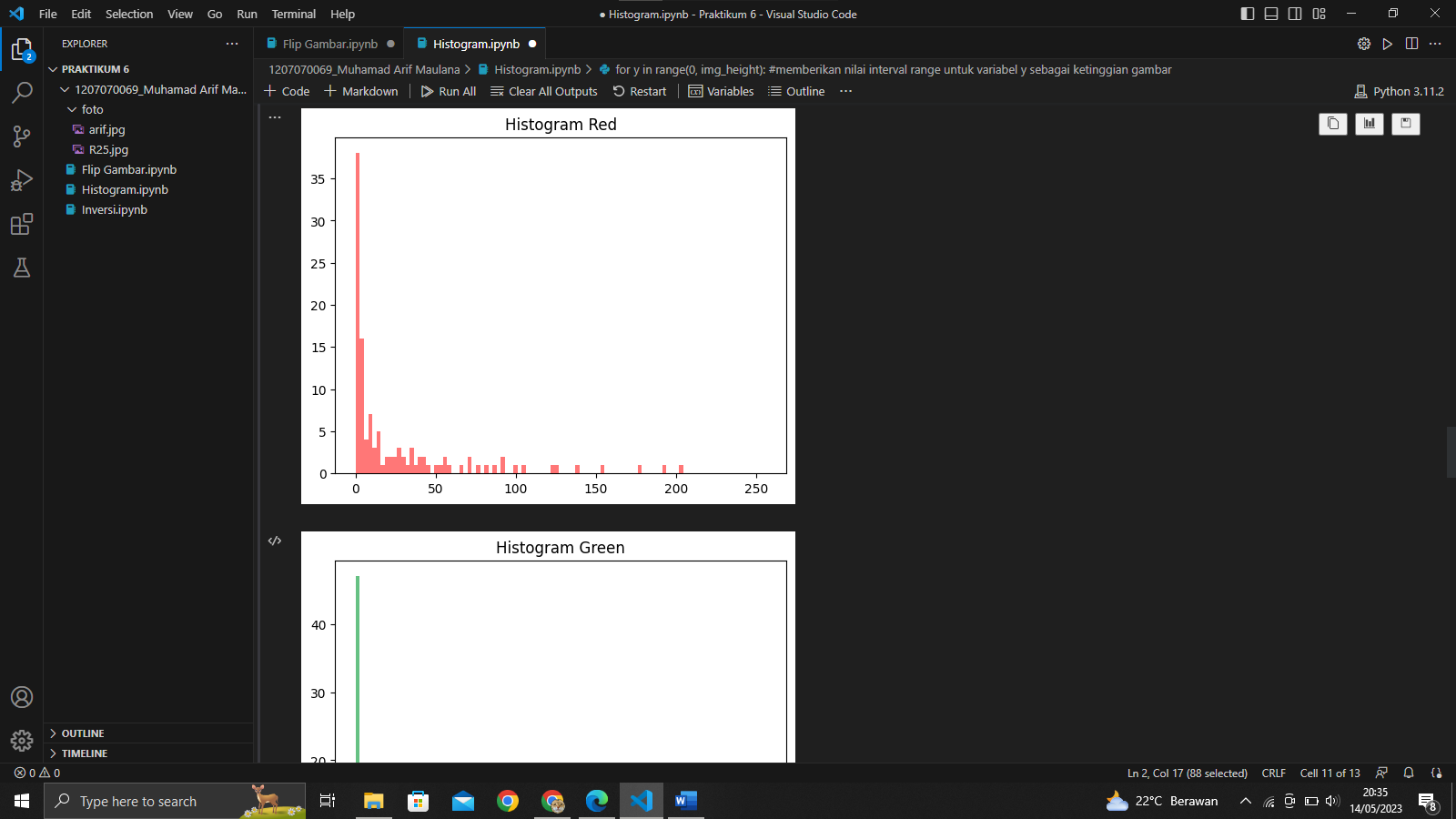
plt.title("Histogram Green") #memberi nama pada gambar

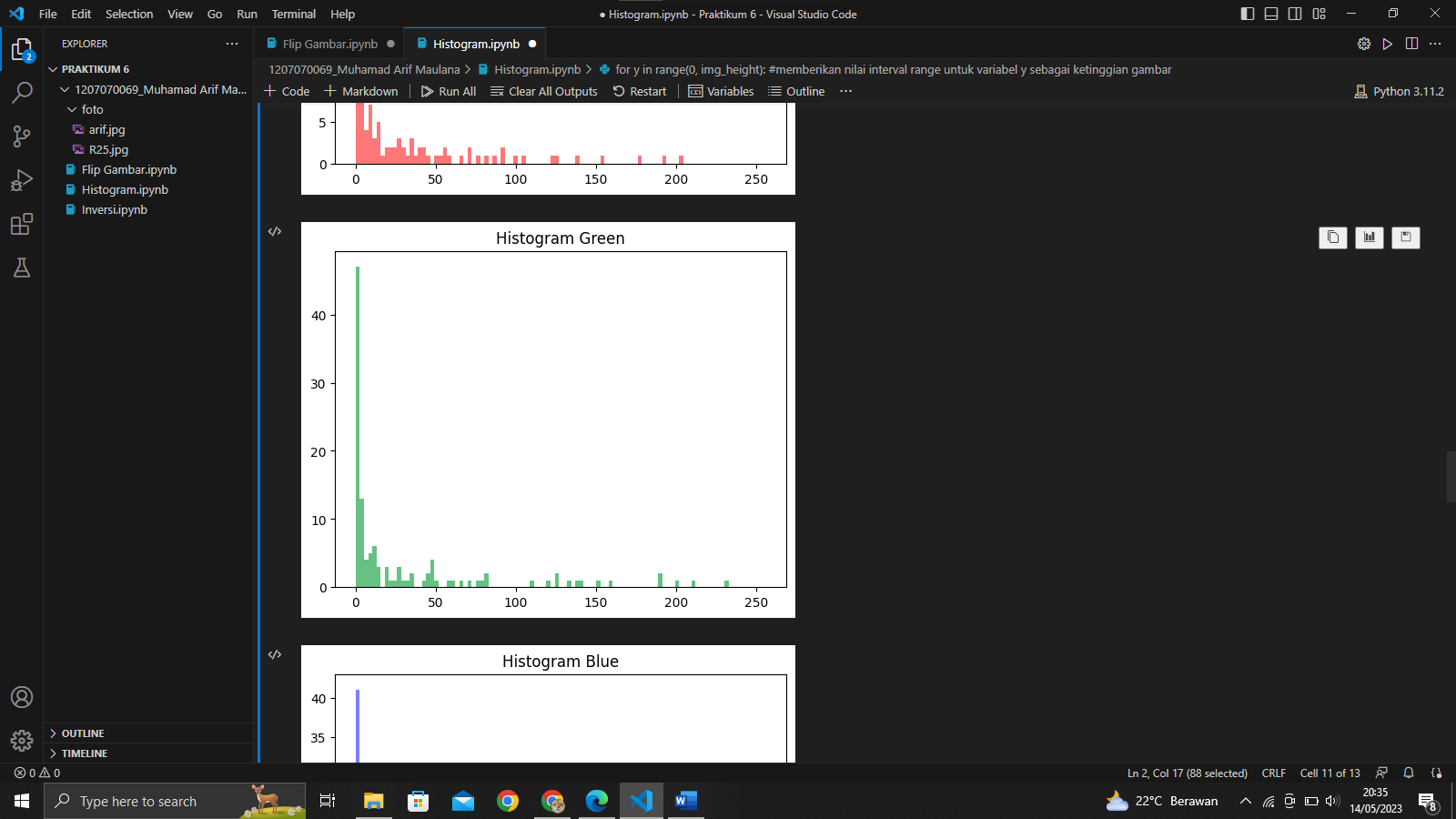
plt.show() #menampilkan keluaran hasil gambar

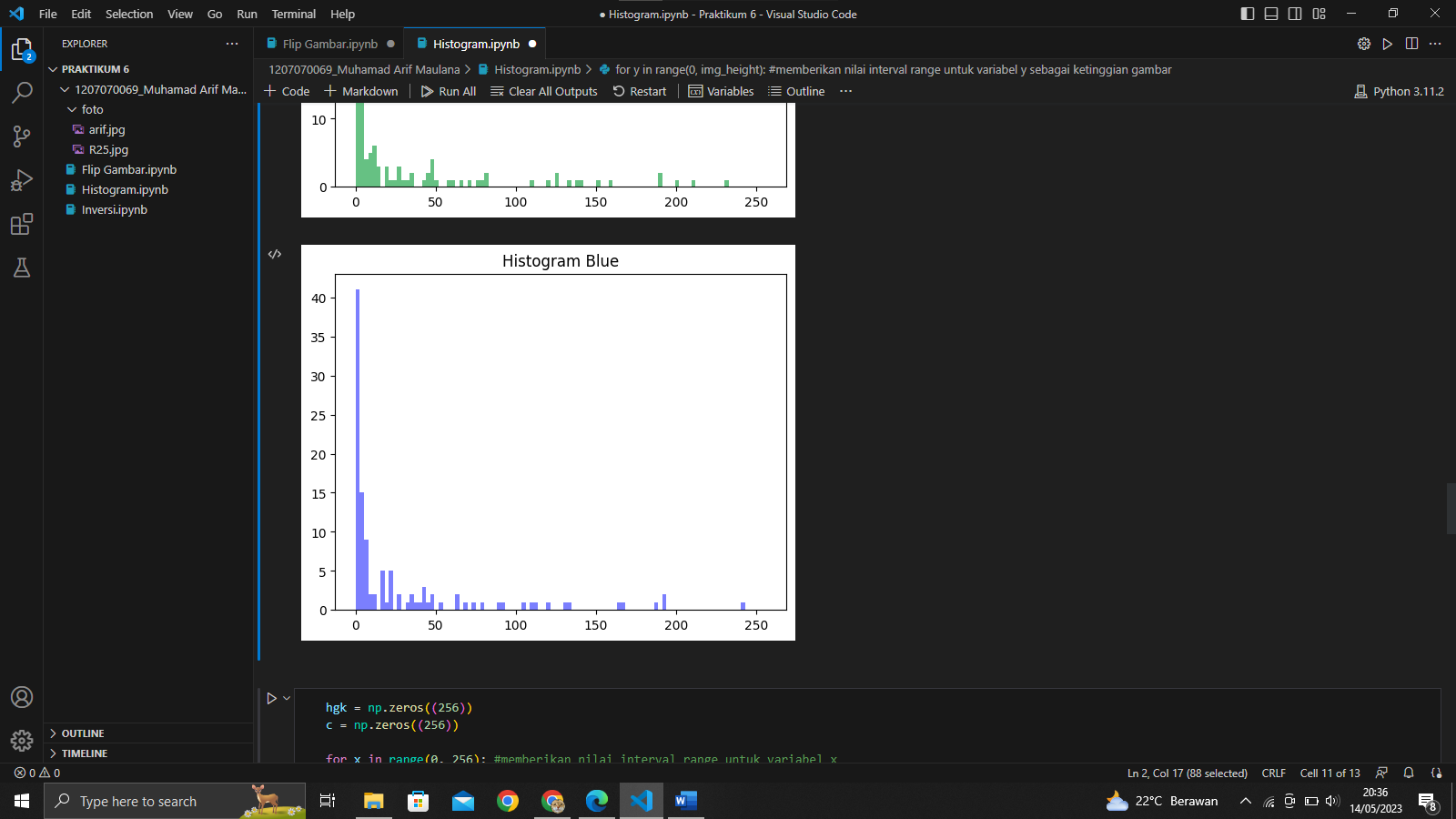
plt.hist(hgb, bins, color="blue", alpha=0.5) #menampilkan histogram dengan warna biru saja pada histogram

plt.title("Histogram Blue") #memberi nama pada gambar

plt.show() #menampilkan keluaran hasil gambar







hgk = np.zeros((256))

c = np.zeros((256))

for x in range(0, 256): #memberikan nilai interval range untuk variabel x

    hgk[x] = 0

    c[x] = 0

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 interval range untuk variabel x sebagai lebar gambar

        gray = img\_grayscale[y][x][0] #memberikan nilai pixel untuk grayscale

        hgk[gray] += 1

c[0] = hgk[0]

for x in range(1, 256): #memberikan nilai interval range untuk variabel x

     c[x] = c[x-1] + hgk[x]

hmaxk = c[255]

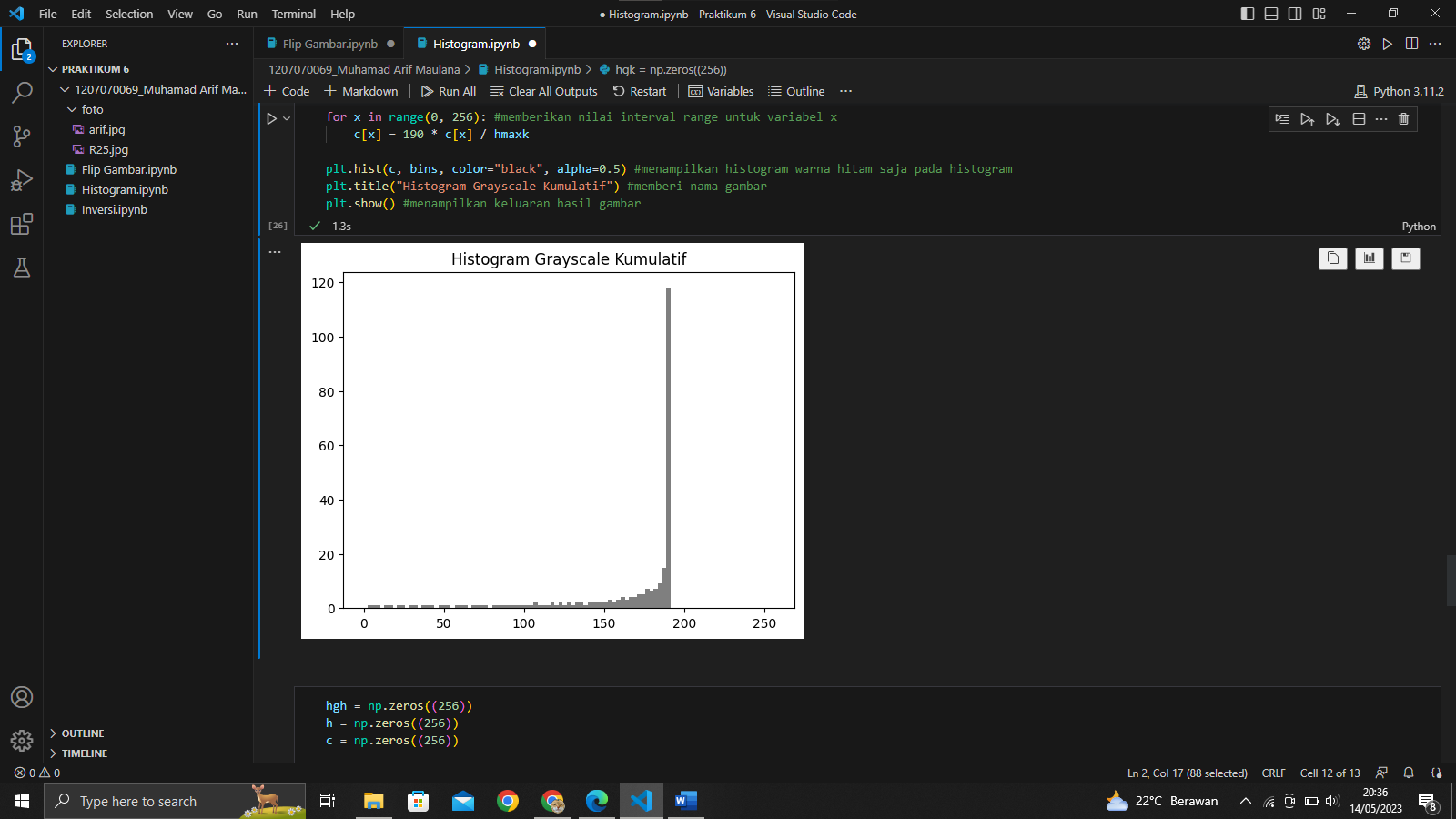
for x in range(0, 256): #memberikan nilai interval range untuk variabel x

    c[x] = 190 \* c[x] / hmaxk

plt.hist(c, bins, color="black", alpha=0.5) #menampilkan histogram warna hitam saja pada histogram

plt.title("Histogram Grayscale Kumulatif") #memberi nama gambar

plt.show() #menampilkan keluaran hasil gambar



hgh = np.zeros((256))

h = np.zeros((256))

c = np.zeros((256))

for x in range(0, 256): #memberikan nilai interval range untuk variabel x

    hgh[x] = 0

    h[x] = 0

    c[x] = 0

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 interval range untuk variabel x sebagai lebar gambar

        gray = img\_grayscale[y][x][0] #memberikan nilai pixel untuk grayscale

        hgh[gray] += 1

h[0] = hgh[0]

for x in range(1, 256): #memberikan nilai interval range untuk variabel x

     h[x] = h[x-1] + hgh[x]

for x in range(0, 256): #memberikan nilai interval range untuk variabel x

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

for x in range(0, 256): #memberikan nilai interval range untuk variabel x

    hgh[x] = 0

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 interval range untuk variabel x sebagai lebar gambar

        gray = img\_grayscale[y][x][0] #memberikan nilai pixel untuk grayscale

        gray = h[gray] \* 255

        hgh[int(gray)] += 1

c[0] = hgh[0]

for x in range(1, 256): #memberikan nilai interval range untuk variabel x

     c[x] = c[x-1] + hgh[x]

hmaxk = c[255]

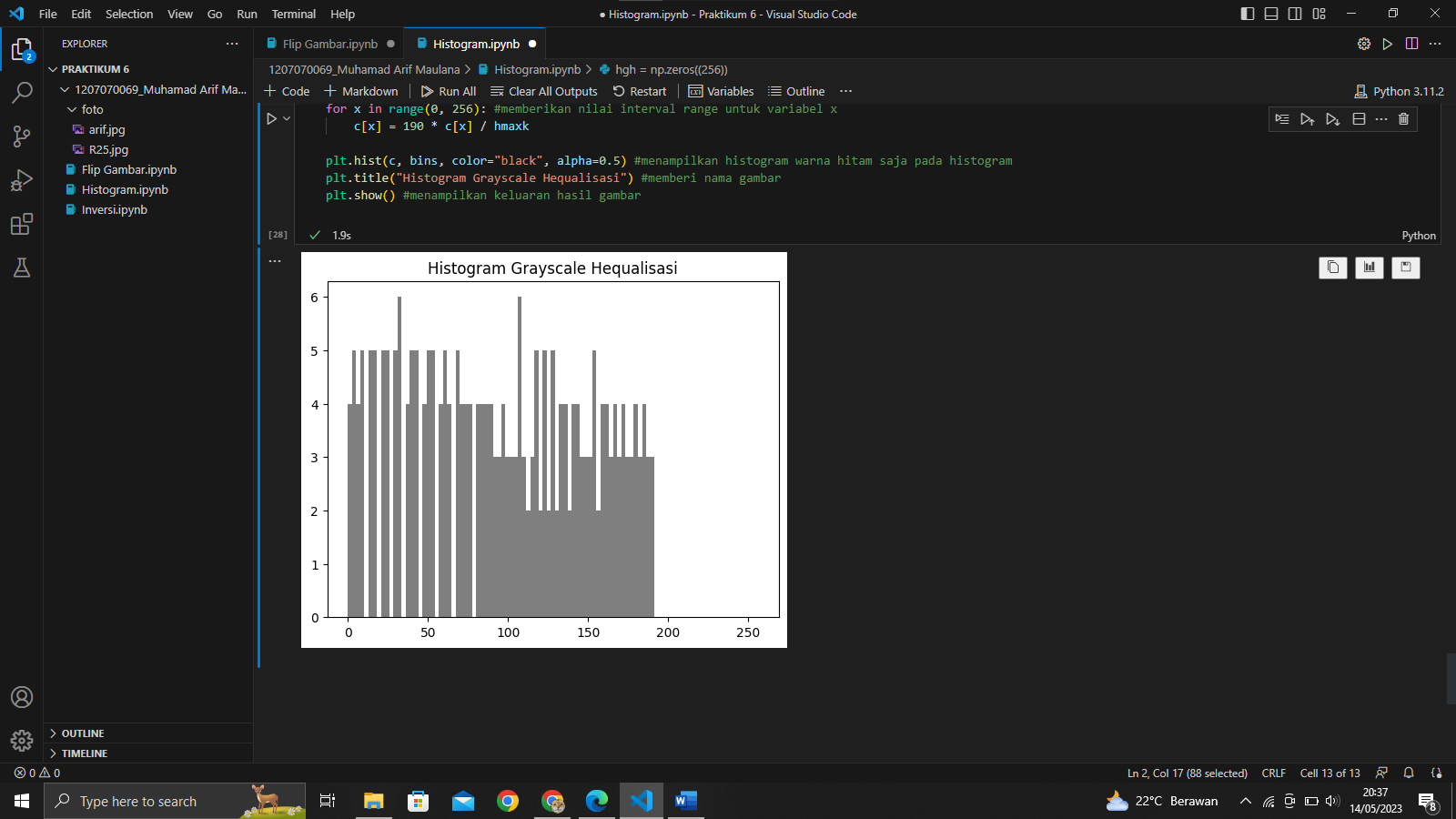
for x in range(0, 256): #memberikan nilai interval range untuk variabel x

    c[x] = 190 \* c[x] / hmaxk

plt.hist(c, bins, color="black", alpha=0.5) #menampilkan histogram warna hitam saja pada histogram

plt.title("Histogram Grayscale Hequalisasi") #memberi nama gambar

plt.show() #menampilkan keluaran hasil gambar



**Inversi**

Source Code

import numpy as np #memanggilan library numpy

import cv2 #memanggil library cv2

import matplotlib.pyplot as plt  #memanggil library matplotlib

img = cv2.imread("foto/arif.jpg") #memanggil gambar yang terdapat pada folder

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

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

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

img\_inversi = np.zeros(img.shape, dtype=np.uint8) #mengosongkan aray untuk menyimpan pixel

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) #menampilkan hasil dari perubahan

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") #memberi nama gambar

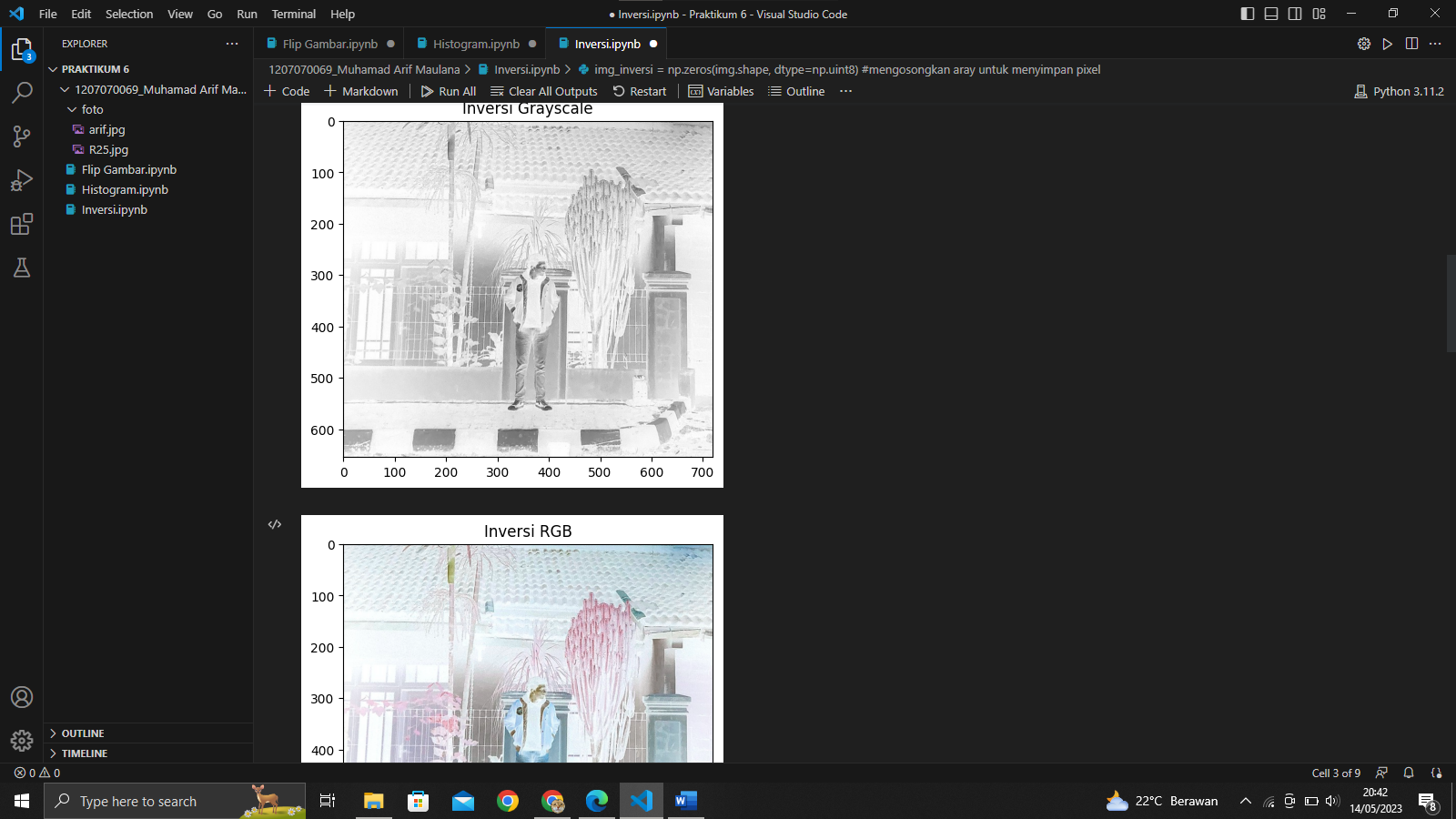
plt.show() #menampilkan hasil keluaran gambar

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

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

plt.title("Inversi RGB") #memberi nama gambar

plt.show() #menampilkan hasil keluaran gambar



img\_inlog = np.zeros(img.shape, dtype=np.uint8) #mengosongkan aray untuk menyimpan pixel

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 pixel 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

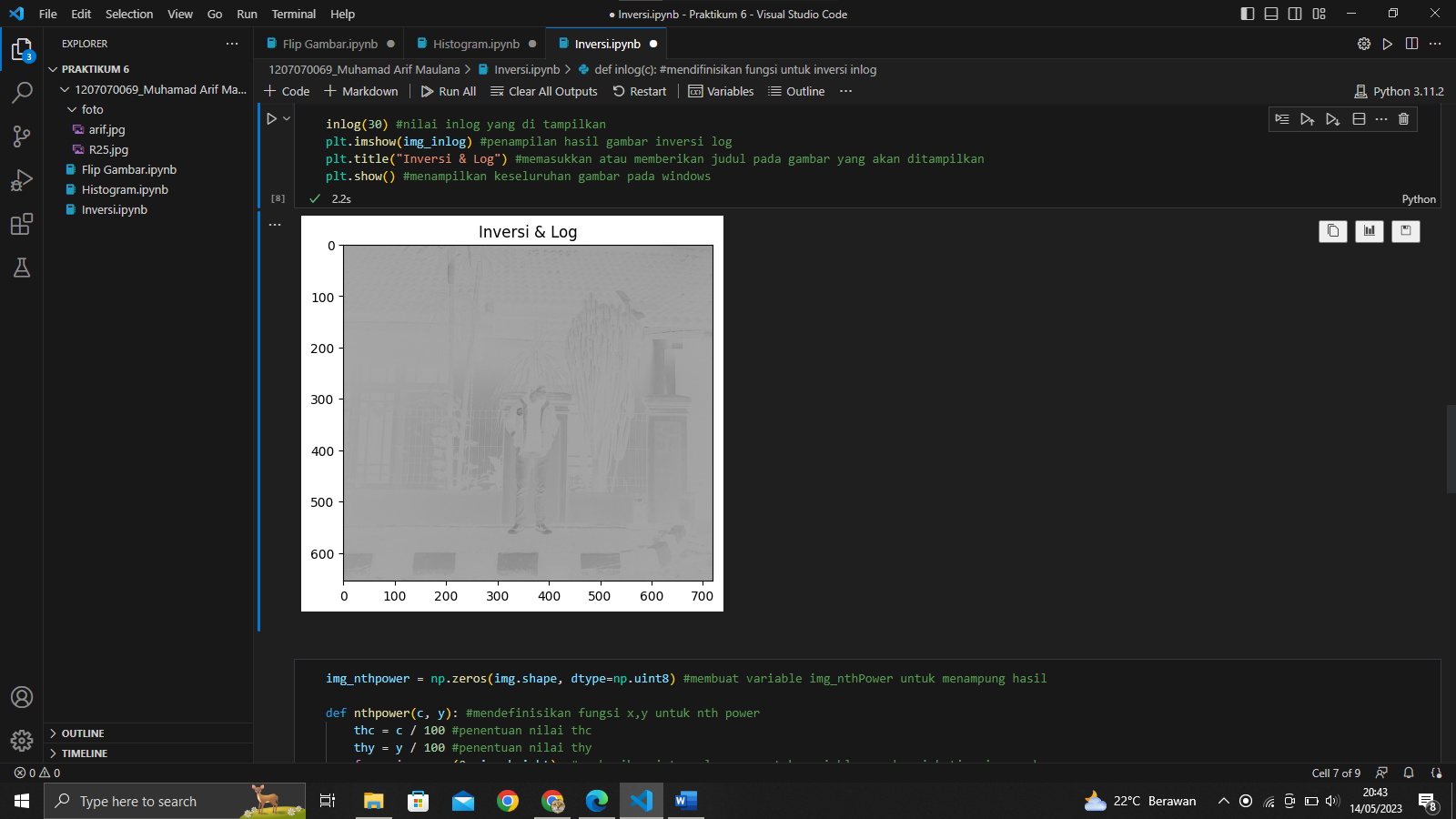
            img\_inlog[y][x] = (gray, gray, gray) #hasil yang keluar sebagai output

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 pixel 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

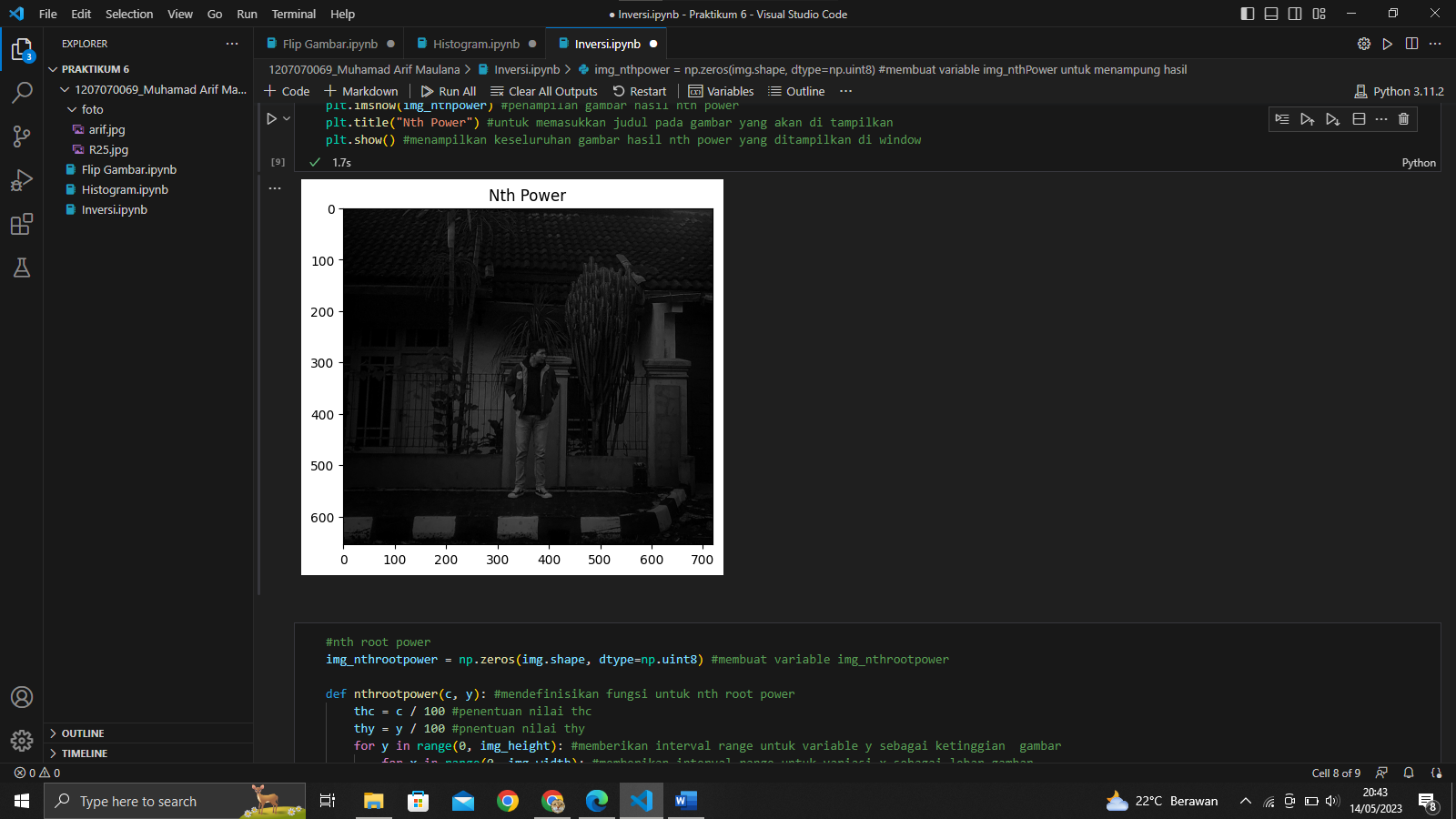
            img\_nthpower[y][x] = (gray, gray, gray) #hasil gambar yang akan ditampilkan

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

