Pengolahan Citra

**Tugas 2 (Ekstraksi Warna)**

**Dosen Pengampu**

Hero Yudo Martono ST, MT



**Disusun Oleh :**

Nama : M. Faza Nur Husain

Nrp : 3121550004

**D3 PJJ AK TEKNIK INFORMATIKA**

**POLITEKNIK ELEKTRONIKA NEGERI SURABAYA**

**TAHUN AKADEMIK 2021/2022**

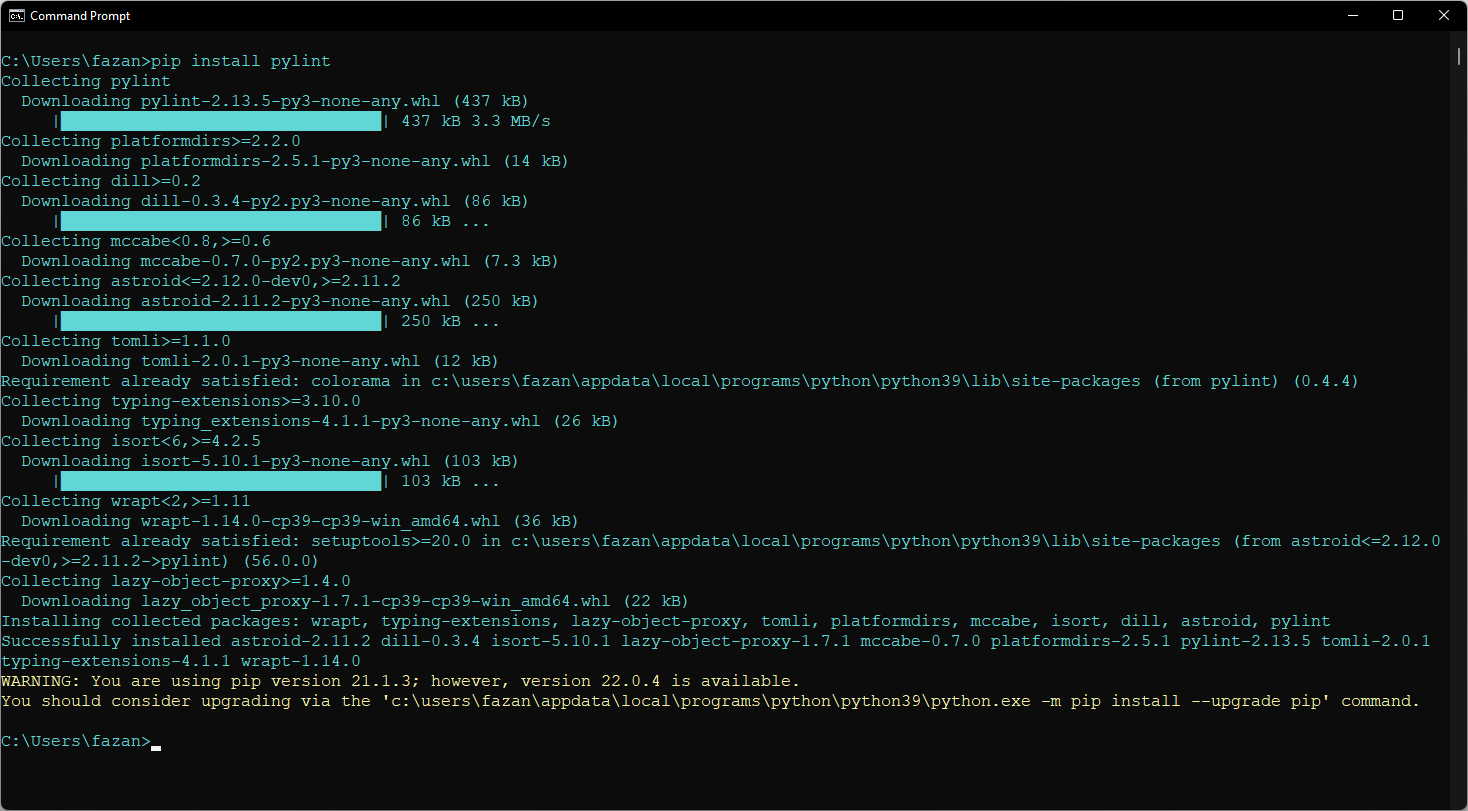
**Membuat aplikasi untuk membedakan bunga berdasarkan warna nya, sertakan script python, jelaskan maksudnya dan capture hasil nya …………**

Langkah 1: Menentukan Deskriptor Gambar kami

Install Paket imutils di Python



Install Paket pylint di Python



Buat dan Buka file baru, beri nama colordescriptor.py :

import numpy as np

import cv2

import imutils

class ColorDescriptor:

    def \_\_init\_\_(self, bins):

        self.bins = bins

    def describe(self, image):

        image = cv2.cvtColor(image, cv2.COLOR\_BGR2HSV)

        features = []

        (h, w) = image.shape[:2]

        (cX, cY) = (int(w \* 0.5), int(h \* 0.5))

        segments = [(0, cX, 0, cY), (cX, w, 0, cY), (cX, w, cY, h),

                    (0, cX, cY, h)]

        (axesX, axesY) = (int(w \* 0.75) // 2, int(h \* 0.75) // 2)

        ellipMask = np.zeros(image.shape[:2], dtype="uint8")

        cv2.ellipse(ellipMask, (cX, cY), (axesX, axesY), 0, 0, 360, 255, -1)

        for (startX, endX, startY, endY) in segments:

            cornerMask = np.zeros(image.shape[:2], dtype="uint8")

            cv2.rectangle(cornerMask, (startX, startY), (endX, endY), 255, -1)

            cornerMask = cv2.subtract(cornerMask, ellipMask)

            hist = self.histogram(image, cornerMask)

            features.extend(hist)

            hist = self.histogram(image, ellipMask)

            features.extend(hist)

            return features

    def histogram(self, image, mask):

        hist = cv2.calcHist([image], [0, 1, 2], mask, self.bins,

                            [0, 180, 0, 256, 0, 256])

        if imutils.is\_cv2():

            hist = cv2.normalize(hist).flatten()

        else:

            hist = cv2.normalize(hist, hist).flatten()

        return hist

Langkah 2: Mengekstrak Fitur dari Dataset

Buat dan Buka file baru, beri nama index.py

from colordescriptor import ColorDescriptor

import argparse

import glob

import cv2

ap = argparse.ArgumentParser()

ap.add\_argument("-d", "--dataset", required=True,

                help="Path to the directory that contains the images to be indexed")

ap.add\_argument("-i", "--index", required=True,

                help="Path to where the computed index will be stored")

args = vars(ap.parse\_args())

cd = ColorDescriptor((8, 12, 3))

output = open(args["index"], "w")

for imagePath in glob.glob(args["dataset"] + "/\*.png"):

    imageID = imagePath[imagePath.rfind("/") + 1:]

    image = cv2.imread(imagePath)

    features = cd.describe(image)

    features = [str(f) for f in features]

    output.write("%s,%s\n" % (imageID, ",".join(features)))

output.close()

Langkah 3: Pencari

Buat dan Buka file baru, beri nama searcher.py

*# impor paket yang diperlukan*

import numpy as np

import csv

class Searcher:

    def \_\_init\_\_(self, indexPath):

        self.indexPath = indexPath

    def search(self, queryFeatures, limit=10):

        results = {}

        with open(self.indexPath) as f:

            reader = csv.reader(f)

            for row in reader:

                features = [float(x) for x in row[1:]]

                d = self.chi2\_distance(features, queryFeatures)

                results[row[0]] = d

            f.close()

        results = sorted([(v, k) for (k, v) in results.items()])

        return results[:limit]

    def chi2\_distance(self, histA, histB, eps=1e-10):

        d = 0.5 \* np.sum([((a - b) \*\* 2) / (a + b + eps)

                          for (a, b) in zip(histA, histB)])

        return d

Langkah 4: Melakukan Pencarian

Buat dan Buka file baru, beri nama search.py

from colordescriptor import ColorDescriptor

from searcher import Searcher

import argparse

import cv2

ap = argparse.ArgumentParser()

ap.add\_argument("-i", "--index", required=True,

                help="Path to where the computed index will be stored")

ap.add\_argument("-q", "--query", required=True,

                help="Path to the query image")

ap.add\_argument("-r", "--result-path", required=True,

                help="Path to the result path")

args = vars(ap.parse\_args())

cd = ColorDescriptor((8, 12, 3))

query = cv2.imread(args["query"])

features = cd.describe(query)

searcher = Searcher(args["index"])

results = searcher.search(features)

cv2.imshow("Query", query)

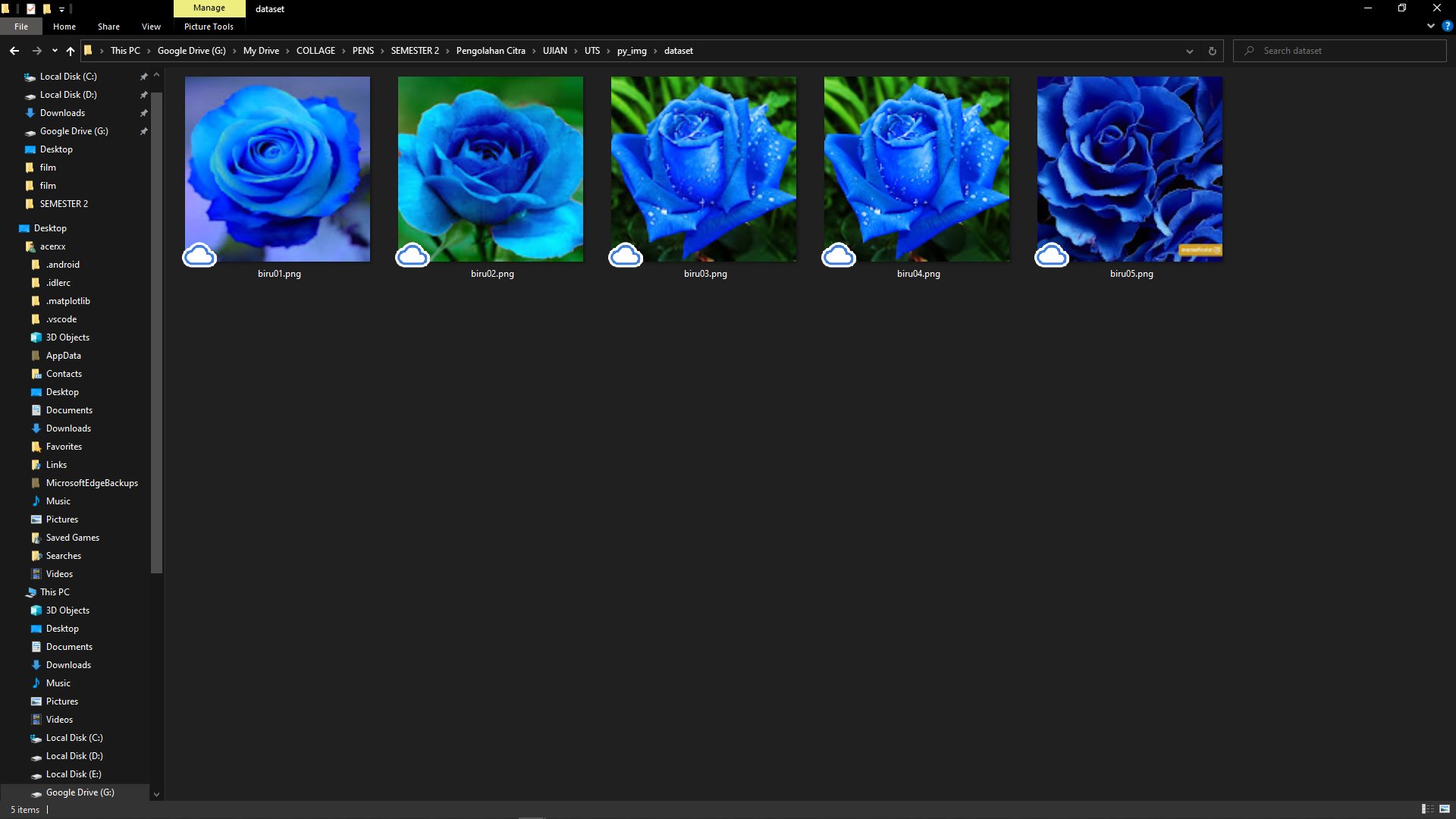
for (score, resultID) in results:

    result = cv2.imread(resultID)

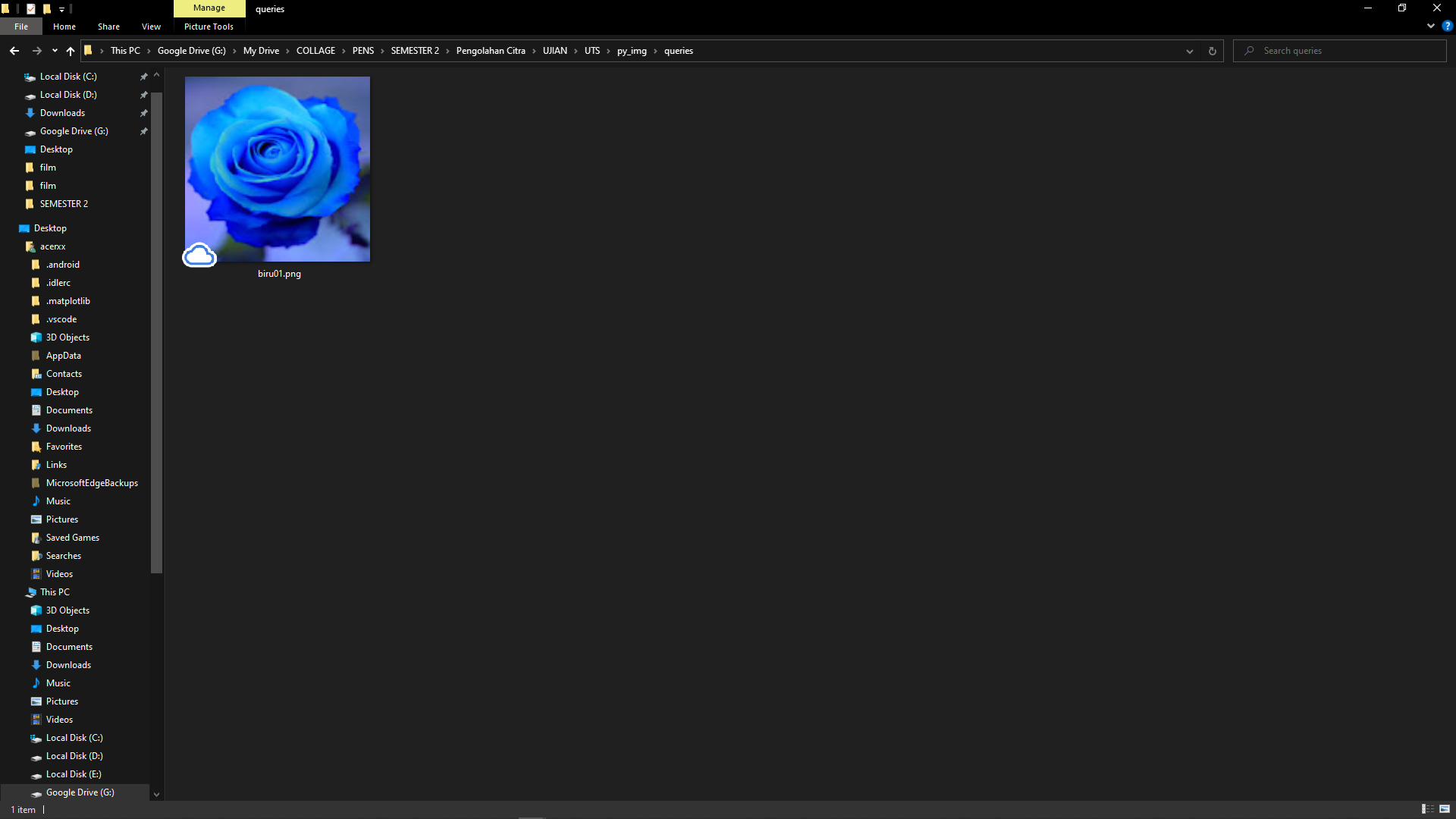
    cv2.imshow("Result", result)

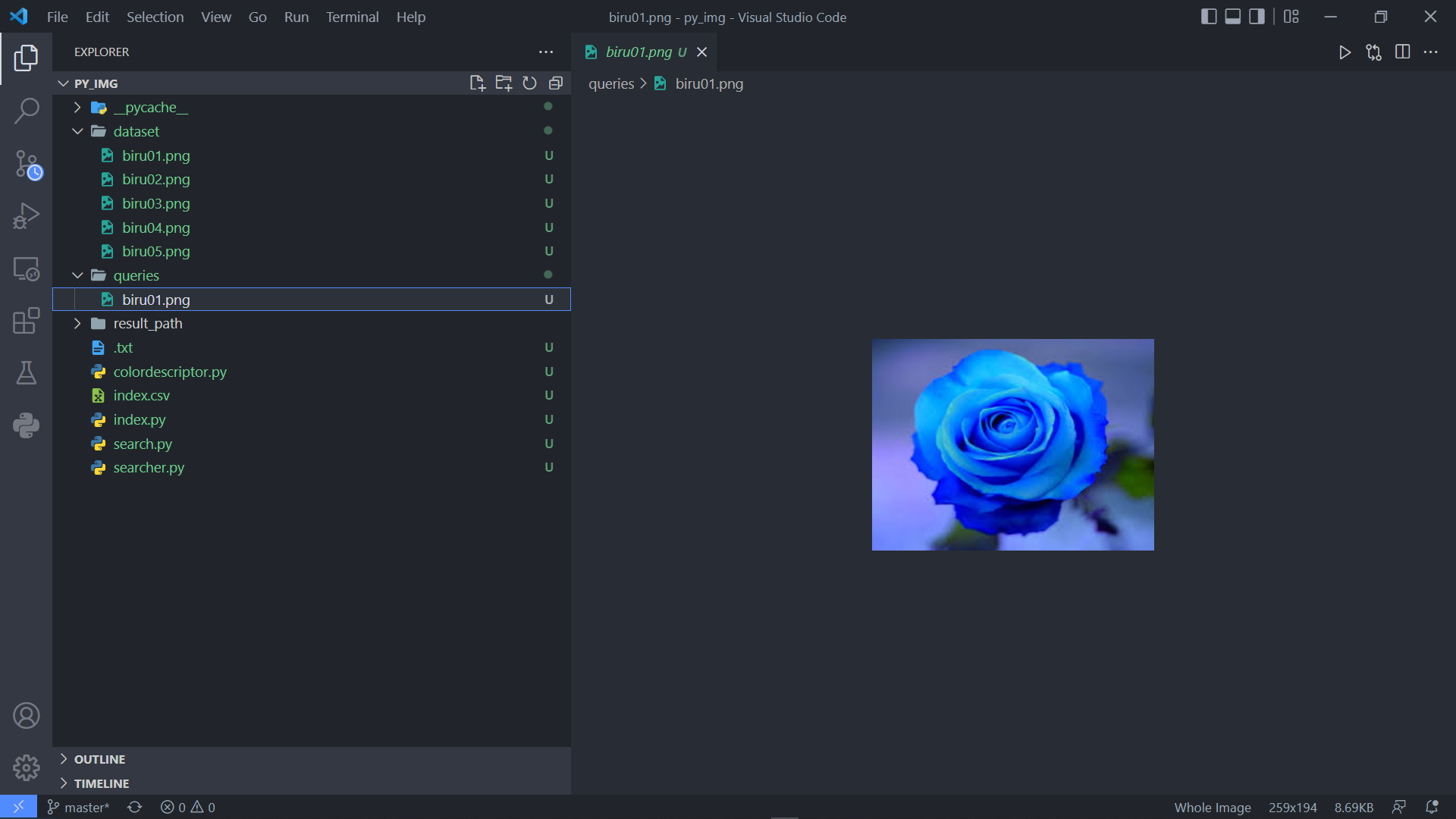
    cv2.waitKey(0)

Buat foder dataset, dan isi gambar



Buat folder queries, dan isi gambar



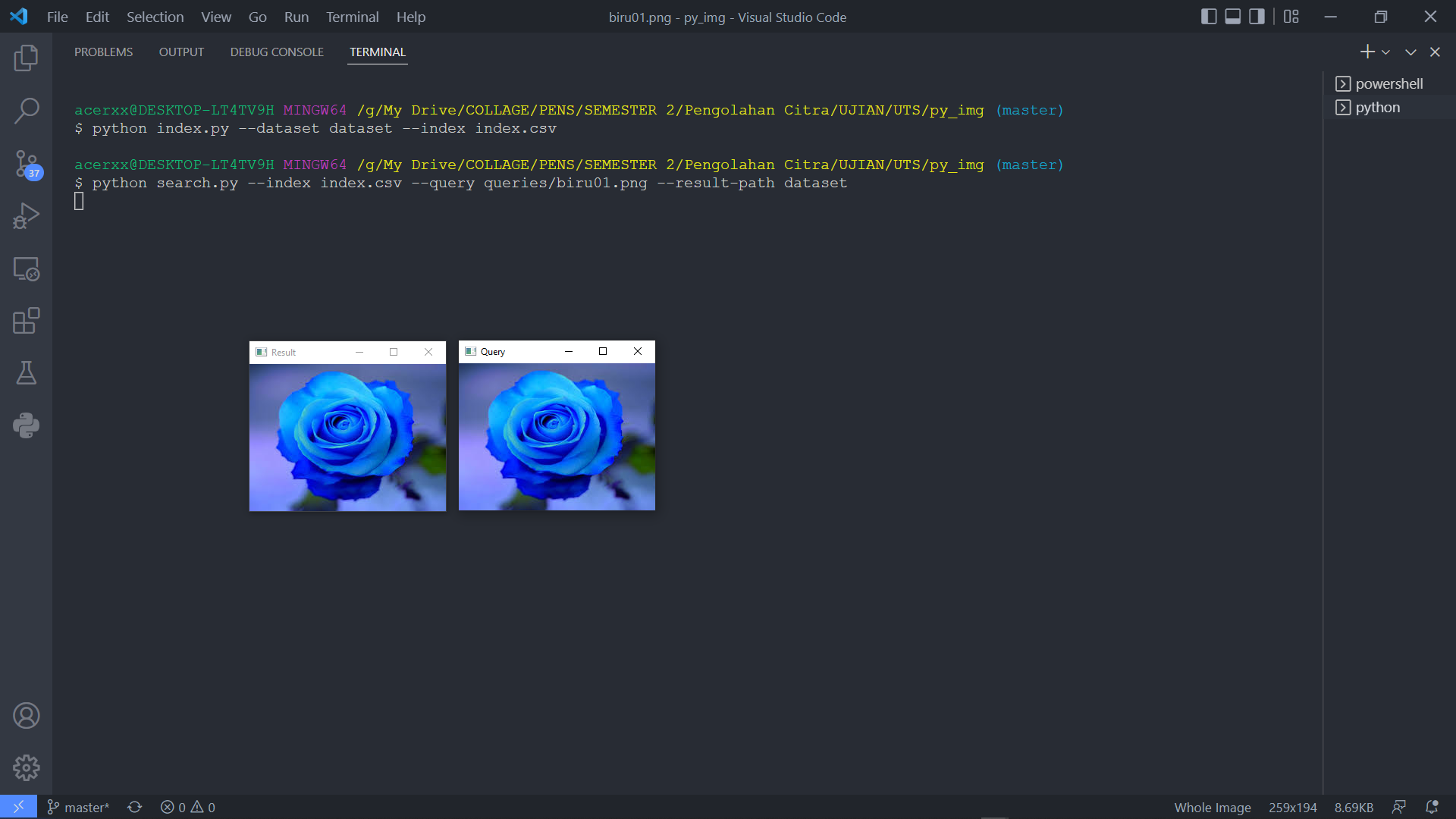


Untuk menjalankan program buka terminal, navigasikan ke direktori tempat program berada, dan jalankan perintah berikut:

python index.py --dataset dataset --index index.csv

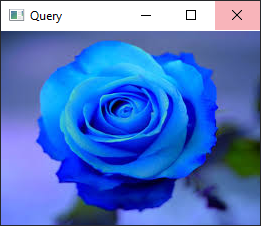
kemudian

python search.py --index index.csv --query queries/biru01.png --result-path dataset



Hasil dari program

Query



Result

