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

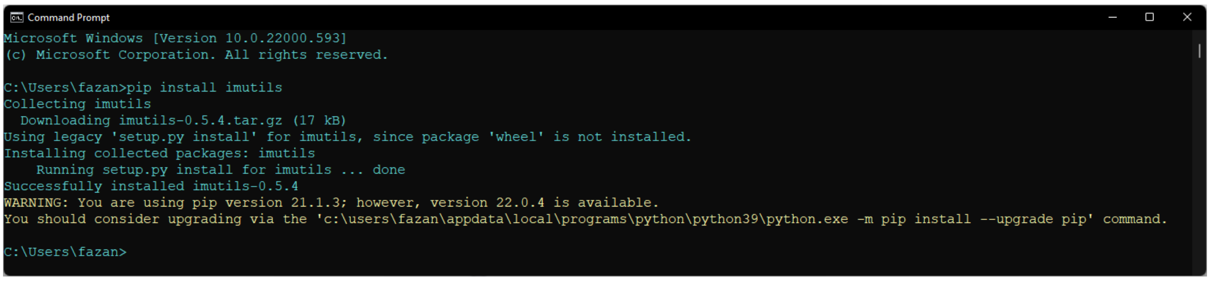
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

TAHUN AKADEMIK 2021/2022

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

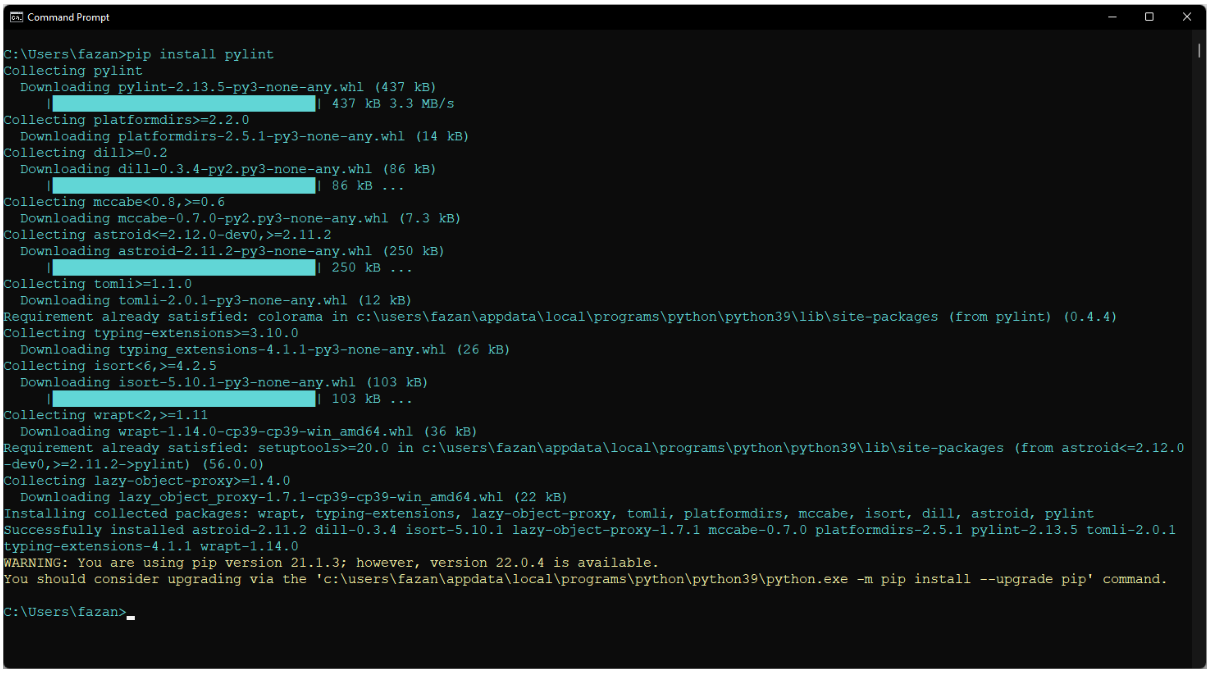
Langkah 1: Menentukan Deskriptor Gambar kami

Install Paket imutils di Python



```
Microsoft Windows [Version 10.0.22000.593]
(c) Microsoft Corporation. All rights reserved.

C:\Users\fazan>pip install imutils
Collecting imutils
  Downloading imutils-0.5.4.tar.gz (17 kB)
Using legacy 'setup.py install' for imutils, since package 'wheel' is not installed.
Installing collected packages: imutils
    Running setup.py install for imutils ... done
Successfully installed imutils-0.5.4
WARNING: You are using pip version 21.1.3; however, version 22.0.4 is available.
You should consider upgrading via the 'c:\users\fazan\appdata\local\programs\python\python39\python.exe -m pip install --upgrade pip' command.

C:\Users\fazan>
```

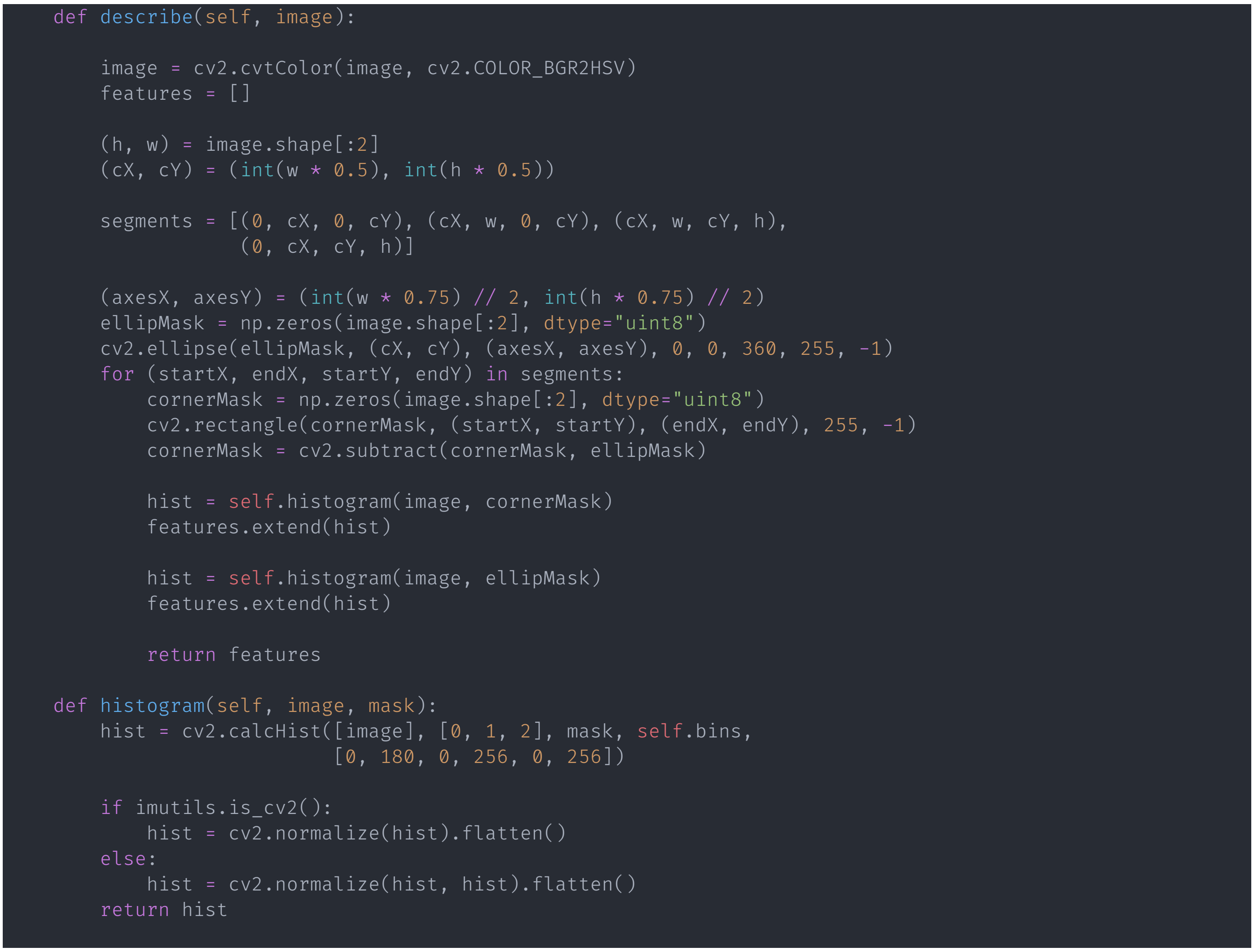
Install Paket pylint di Python



```
C:\Users\fazan>pip install pylint
Collecting pylint
  Downloading pylint-2.13.5-py3-none-any.whl (437 kB)
     |████████████████████████████████| 437 kB 3.3 MB/s
Collecting platformdirs>=2.2.0
  Downloading platformdirs-2.5.1-py3-none-any.whl (14 kB)
Collecting dill>=0.2
  Downloading dill-0.3.4-py2.py3-none-any.whl (86 kB)
     |████████████████████████████████| 86 kB ...
Collecting mccabe<0.8,>=0.6
  Downloading mccabe-0.7.0-py2.py3-none-any.whl (7.3 kB)
Collecting astroid<=2.12.0-dev0,>=2.11.2
  Downloading astroid-2.11.2-py3-none-any.whl (250 kB)
     |████████████████████████████████| 250 kB ...
Collecting tomli>=1.1.0
  Downloading tomli-2.0.1-py3-none-any.whl (12 kB)
Requirement already satisfied: colorama in c:\users\fazan\appdata\local\programs\python\python39\lib\site-packages (from pylint) (0.4.4)
Collecting typing-extensions>=3.10.0
  Downloading typing_extensions-4.1.1-py3-none-any.whl (26 kB)
Collecting isort<6,>=4.2.5
  Downloading isort-5.10.1-py3-none-any.whl (103 kB)
     |████████████████████████████████| 103 kB ...
Collecting wrapt<2,>=1.11
  Downloading wrapt-1.14.0-cp39-cp39-win_amd64.whl (36 kB)
Requirement already satisfied: setuptools>=20.0 in c:\users\fazan\appdata\local\programs\python\python39\lib\site-packages (from astroid<=2.12.0
-dev0,>=2.11.2->pylint) (56.0.0)
Collecting lazy-object-proxy>=1.4.0
  Downloading lazy_object_proxy-1.7.1-cp39-cp39-win_amd64.whl (22 kB)
Installing collected packages: wrapt, typing-extensions, lazy-object-proxy, tomli, platformdirs, mccabe, isort, dill, astroid, pylint
Successfully installed astroid-2.11.2 dill-0.3.4 isort-5.10.1 lazy-object-proxy-1.7.1 mccabe-0.7.0 platformdirs-2.5.1 pylint-2.13.5 tomli-2.0.1
typing-extensions-4.1.1 wrapt-1.14.0
WARNING: You are using pip version 21.1.3; however, version 22.0.4 is available.
You should consider upgrading via the 'c:\users\fazan\appdata\local\programs\python\python39\python.exe -m pip install --upgrade pip' command.

C:\Users\fazan>
```

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

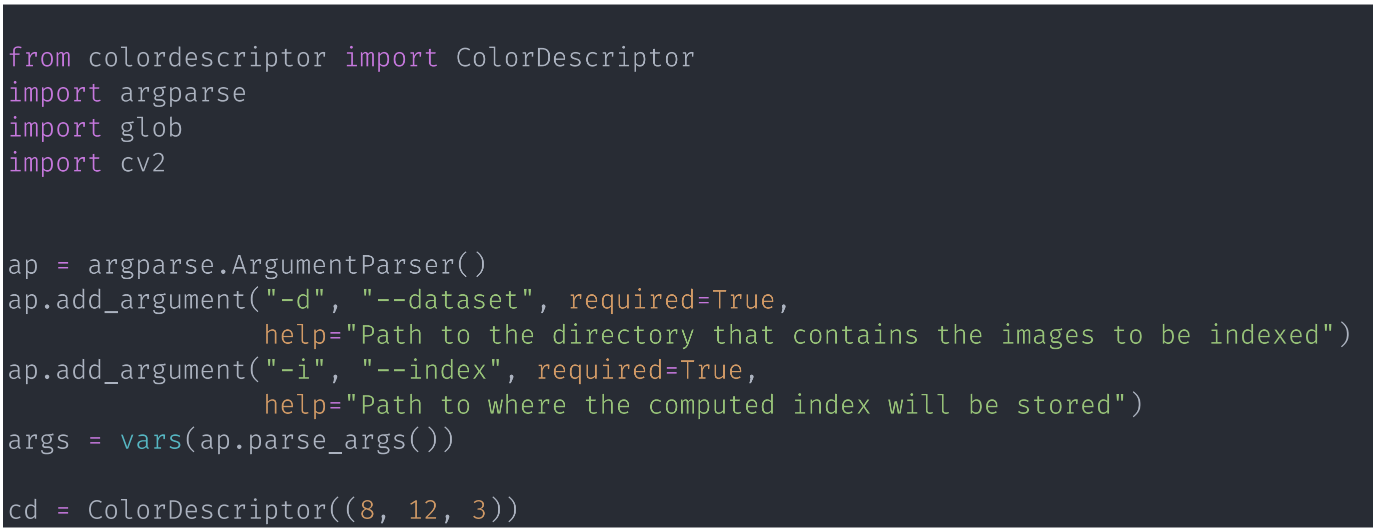
```
import numpy as np
import cv2
import imutils


class ColorDescriptor:
    def __init__(self, bins):
        self.bins = bins
```



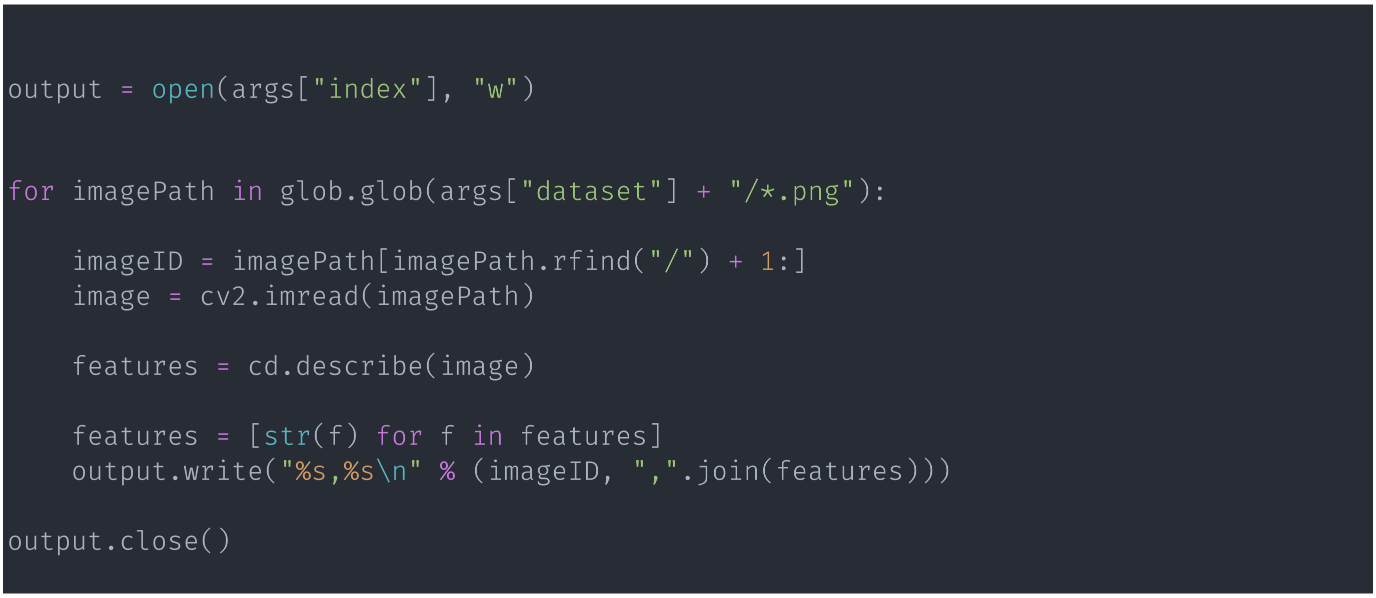
```
    def describe(self, image):

        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



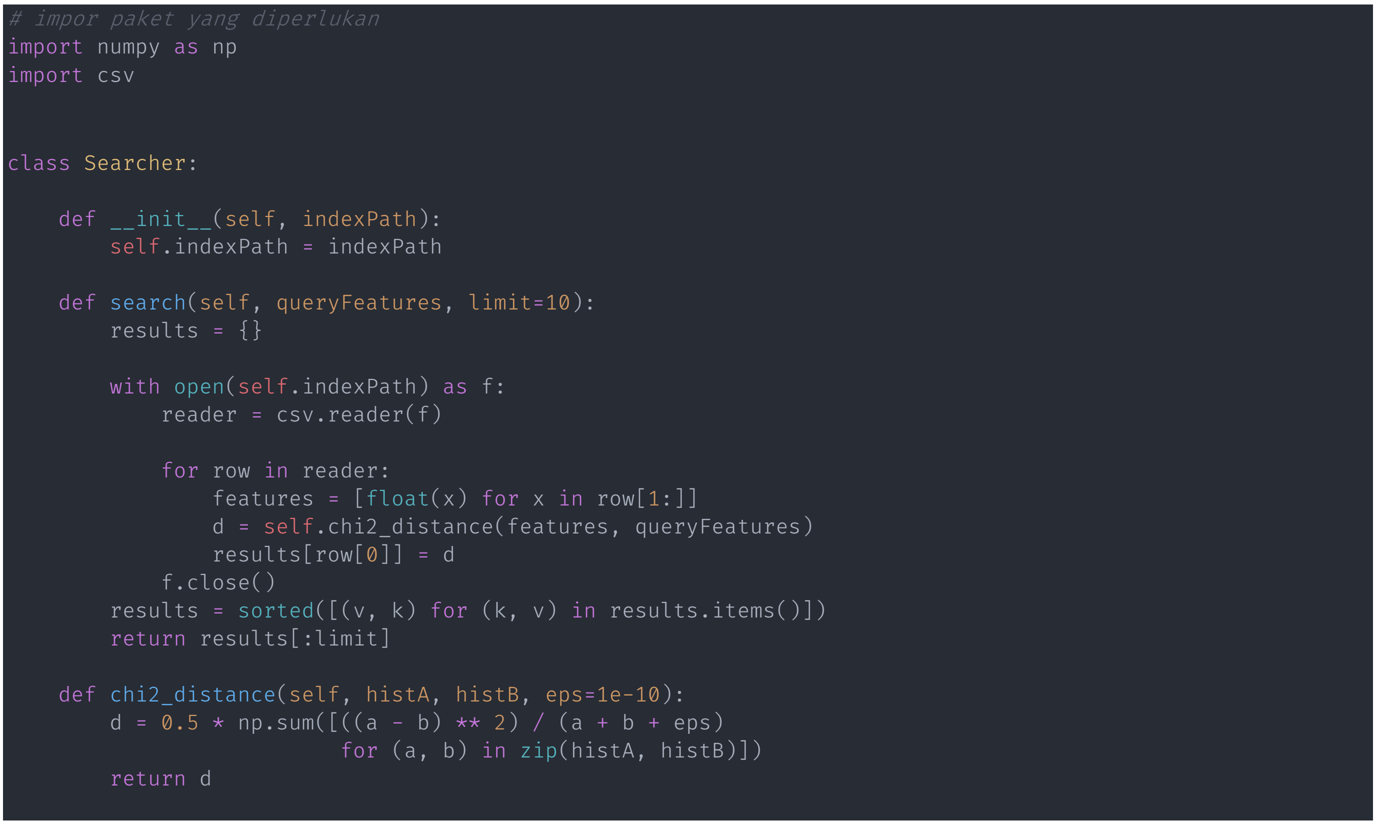
```
from colordescriptor import ColorDescriptor
import argparse
import glob
import cv2


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

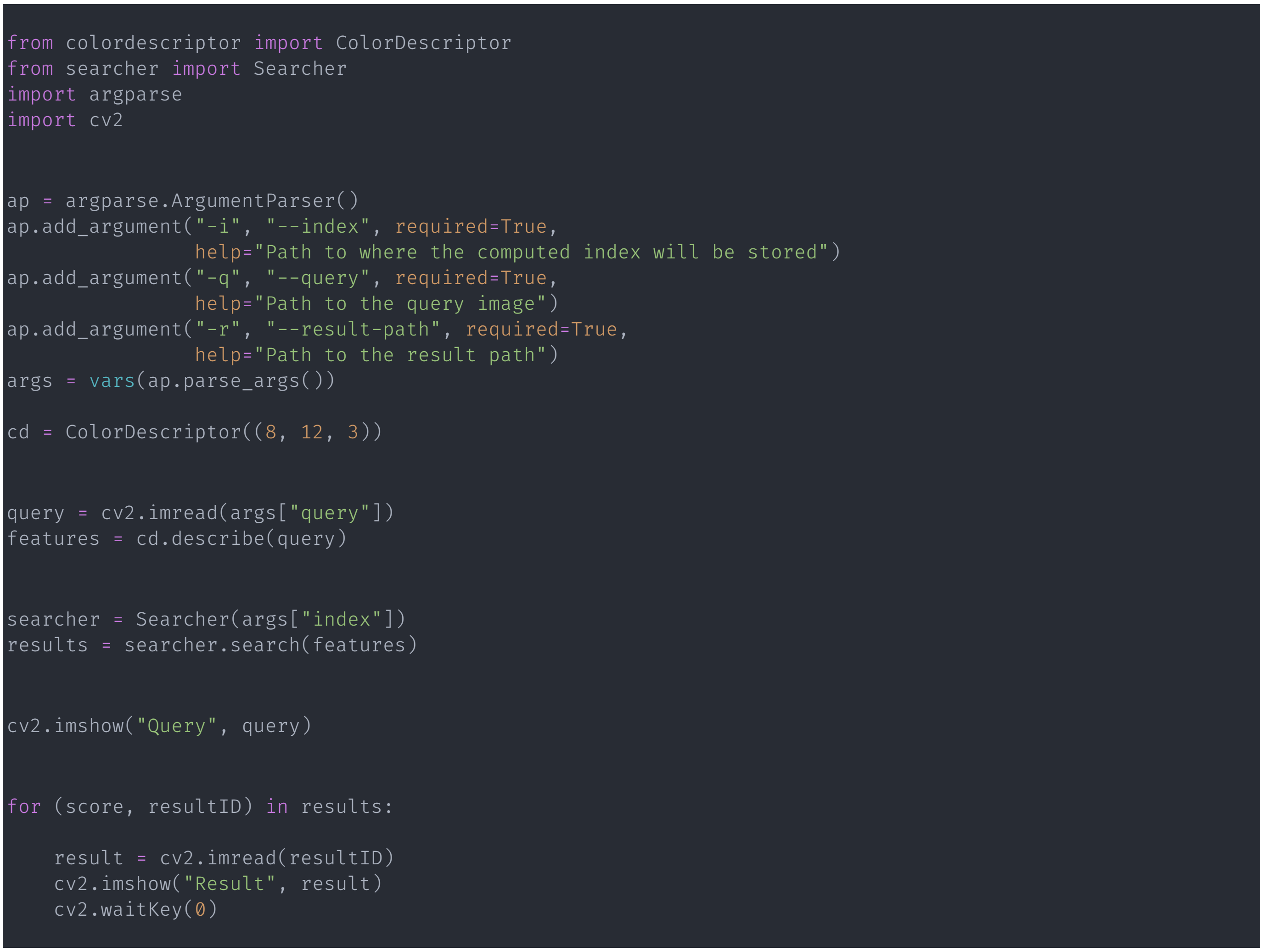
Buat dan Buka file baru, beri nama searcher.py



```
# 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
```

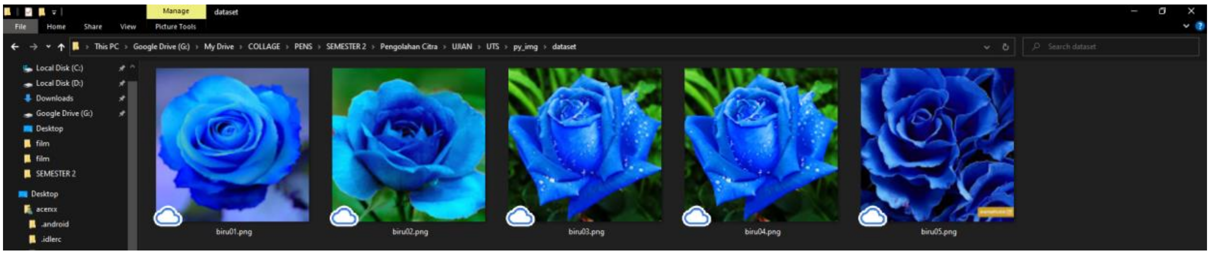
Langkah 4: Melakukan Pencarian

Buat dan Buka file baru, beri nama search.py

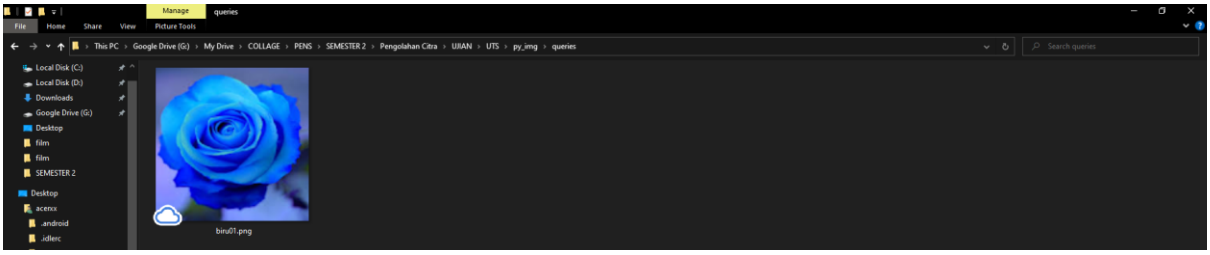


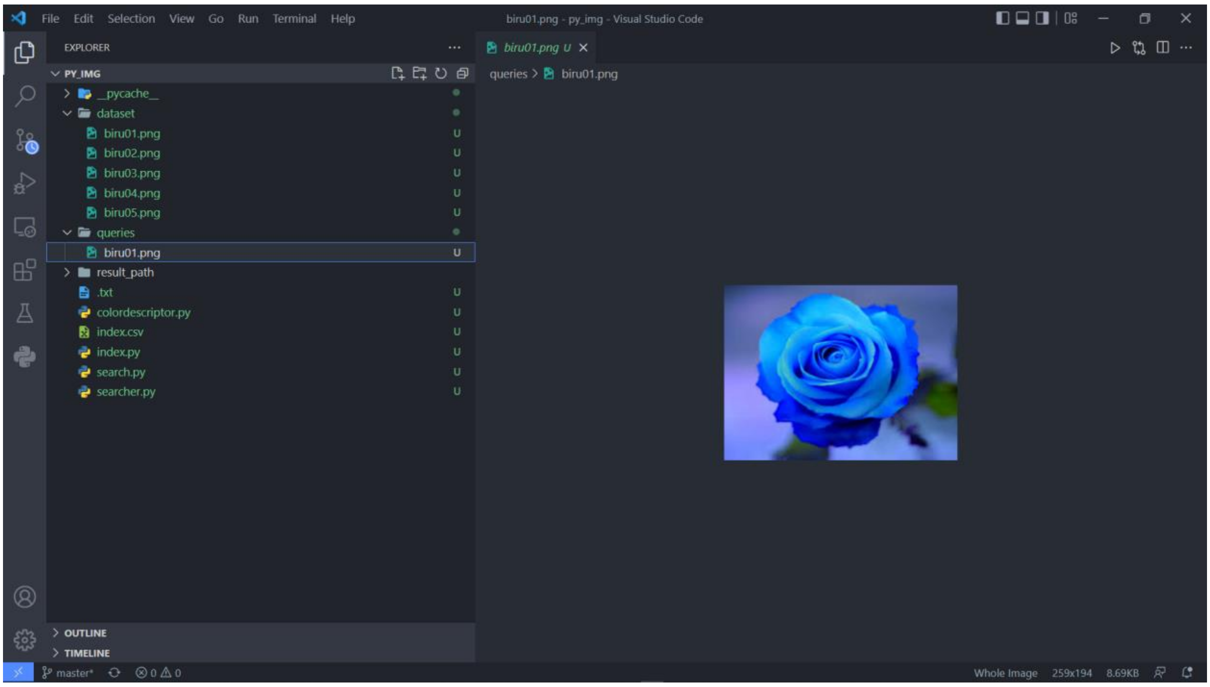
```python
from colordescriptor import ColorDescriptor
from searcher import Searcher
import argparse
import cv2


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



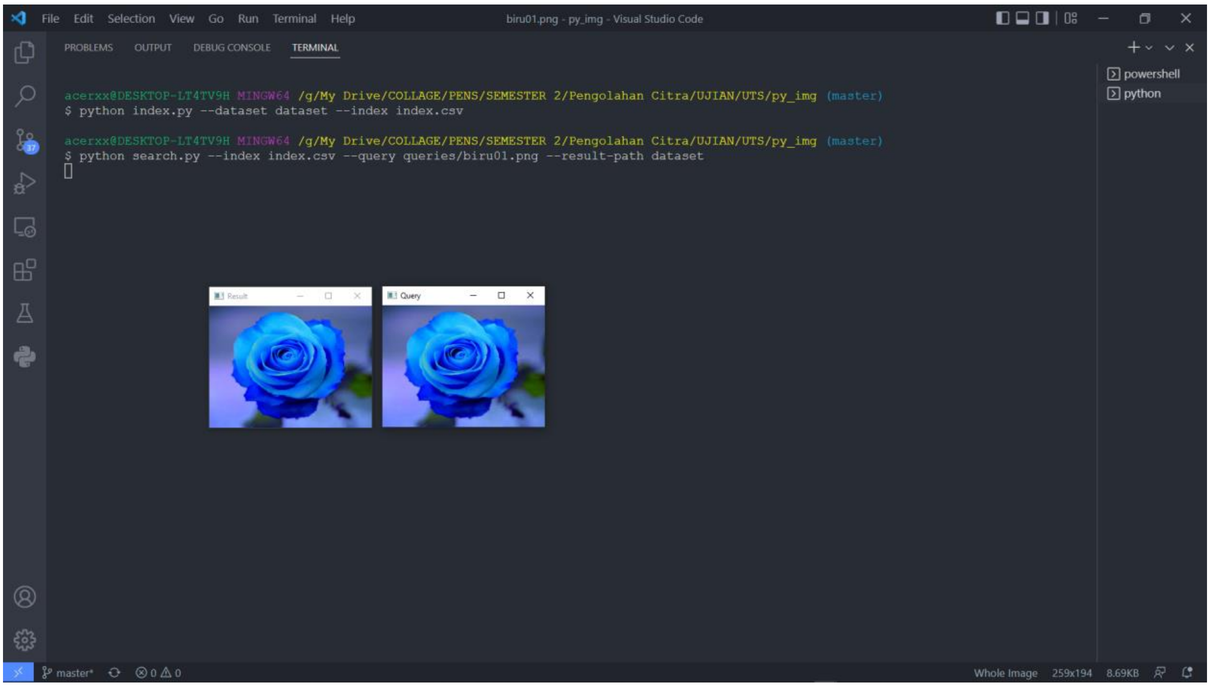


Untuk menjalankan program buka terminal, navigasikan ke direktori tempat program berada, dan jalankan perintah berikut:

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

kemudian

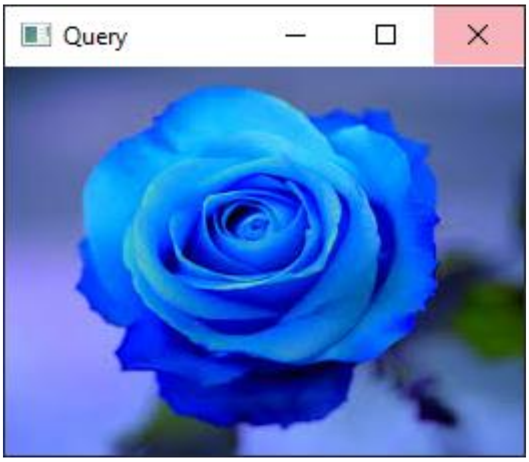
python search.py --index index.csv --query queries/biru01.png --result-path dataset



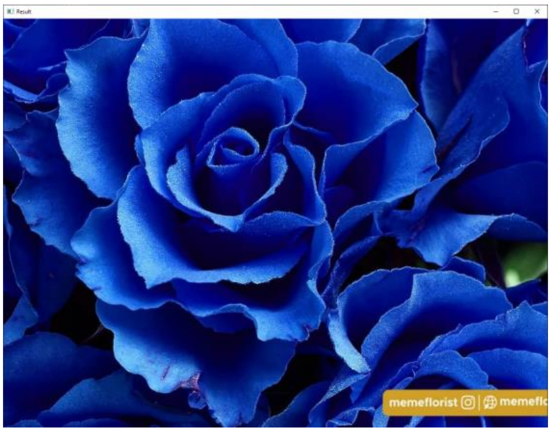
```
acerxx@DESKTOP-LT4TV9H MINGW64 /g/My Drive/COLLAGE/PENS/SEMESTER 2/Pengolahan Citra/UJIAN/UTS/py_img (master)
$ python index.py --dataset dataset --index index.csv

acerxx@DESKTOP-LT4TV9H MINGW64 /g/My Drive/COLLAGE/PENS/SEMESTER 2/Pengolahan Citra/UJIAN/UTS/py_img (master)
$ python search.py --index index.csv --query queries/biru01.png --result-path dataset
```

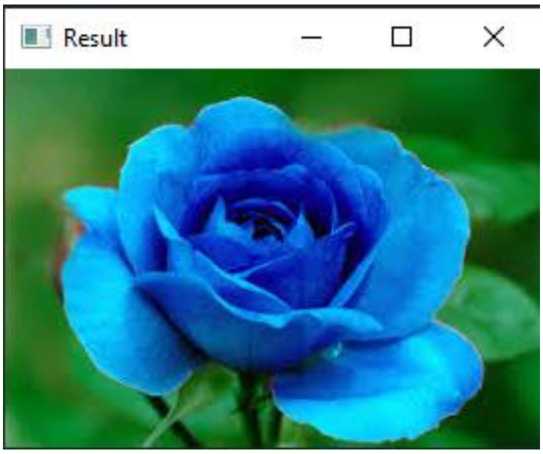
Hasil dari program

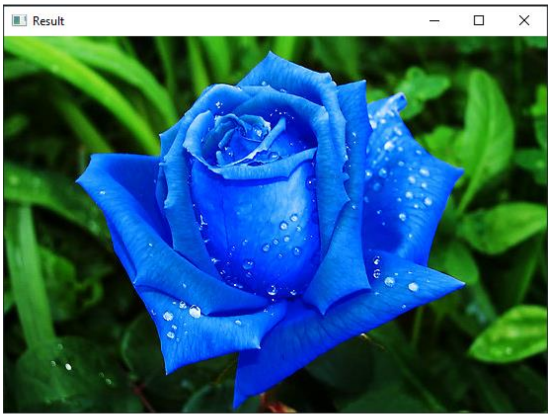
Query

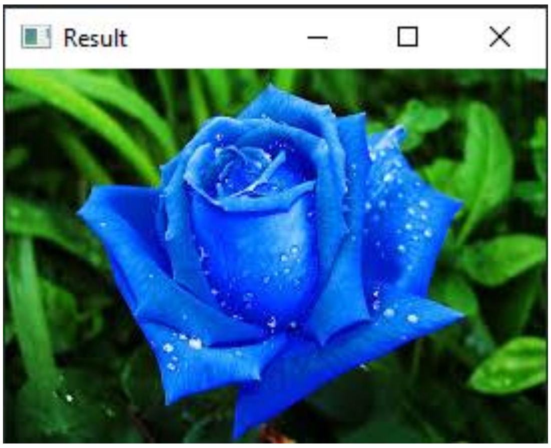


Result






Result


Result