TUGAS PEMROSESAN PARALEL
EKSEKUSI PROGRAM STITCHING MENGGUNAKAN CMD PADA WINDOWS

Disusun untuk memenuhi tugas Mata Kuliah Pemrosesan Paralel

Disusun Oleh:
Rizky Ramadhan
(09011282227087)

Dosen Pengampu:
Adi Hermansyah, S.Kom., M.T.

PROGRAM STUDI SISTEM KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA
PALEMBANG

2023

Hal yang perlu dipersiapkan

1. Windows
2. Cmd
3. Teks editor
4. Coding stitch

Konfigurasi Python

Pastikan bahwa python sudah terinstall di windows.

```
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Acer 4752>python --version
Python 3.11.4
```

Install numpy, imutils, dan opencv

Sebelum menginstall ketiga hal tersebut, kita harus menginstall pip terlebih dahulu

```
C:\Users\Acer 4752>pip --version
pip 23.3.1 from C:\Users\Acer 4752\AppData\Local\Programs\Python\Python311\Lib\site-packages\pip (python 3.11)
```

Lalu install numpy dengan command berikut

```
C:\Users\Acer 4752>pip install numpy
Requirement already satisfied: numpy in c:
26.1)
```

Lalu install imutils dengan command berikut

```
C:\Users\Acer 4752>pip install imutils
Requirement already satisfied: imutils in
0.5.4)
```

Lalu install opencv dengan command berikut

```
C:\Users\Acer 4752>pip install opencv-python
Requirement already satisfied: opencv-python in
ages (4.8.1.78)
Requirement already satisfied: numpy>=1.21.2 in
ages (from opencv-python) (1.26.1)
```

Menyiapkan gambar untuk di stitch

Disini saya telah menyiapkan 9 gambar yang terletak di direktori scottsdale

Coding stitch menggunakan teks editor

Disini saya menggunakan teks editor visual studio code untuk coding stitchnya

```
# USAGE
# python image_stitching_simple.py --images images/scottsdale --output output.png

# import the necessary packages
from imutils import paths
import argparse
import cv2

# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--images", type=str, required=True,
    help="path to input directory of images to stitch")
ap.add_argument("-o", "--output", type=str, required=True,
    help="path to the output image")
args = vars(ap.parse_args())

# grab the paths to the input images and initialize our images list
print("[INFO] loading images...")
imagePaths = sorted(list(paths.list_images(args["images"])))
images = []

# loop over the image paths, load each one, and add them to our
# images to stitch list
for imagePath in imagePaths:
    image = cv2.imread(imagePath)
    images.append(image)

# initialize OpenCV's image stitcher object and then perform the image
# stitching
print("[INFO] stitching images...")

# Create a Stitcher with a default ORB (feature-based) detector
stitcher = cv2.Stitcher_create(cv2.Stitcher_SCANS)

# Detect keypoints and set camera parameters manually
status, stitched = stitcher.stitch(images)
if status != cv2.Stitcher_OK:
    print("[INFO] Camera parameters adjustment failed. Retrying with manual adjus

    # Manually set camera parameters
    stitcher.setWaveCorrection(True)

    # Retry stitching
    status, stitched = stitcher.stitch(images)

# print additional information
print("[INFO] Stitching Status:", status)

# if the status is '0', then OpenCV successfully performed image
# stitching
if status == cv2.Stitcher_OK:
    # write the output stitched image to disk
    cv2.imwrite(args["output"], stitched)

    # display the output stitched image to our screen
    cv2.imshow("Stitched", stitched)
    cv2.waitKey(0)

# otherwise, the stitching failed
else:
    print("[INFO] image stitching failed ({})".format(status))

    # print additional information
    if status == cv2.Stitcher_ERR_NEED_MORE_IMGS:
        print("[INFO] Need more images for stitching.")
    elif status == cv2.Stitcher_ERR_HOMOGRAPHY_EST_FAIL:
        print("[INFO] Homography estimation failed.")
    elif status == cv2.Stitcher_ERR_CAMERA_PARAMS_ADJUST_FAIL:
        print("[INFO] Camera parameters adjustment failed.")
    elif status == cv2.Stitcher_ERR_MATCH_CONFIDENCE_FAIL:
        print("[INFO] Match confidence test failed.")
    elif status == cv2.Stitcher_ERR_CAMERA_PARAMS_VERIFY_FAIL:
        print("[INFO] Camera parameters verification failed.")
```

Run coding di cmd

Pertama pindah dulu ke direktori yang berisi file coding tersebut

```
C:\Users\Acer 4752>cd downloads/imagest/

C:\Users\Acer 4752\Downloads\imagest>
```

Kedua run coding dengan command berikut

```
C:\Users\Acer 4752\Downloads\imagest>python stitch.py
usage: stitch.py [-h] -i IMAGES -o OUTPUT
stitch.py: error: the following arguments are required: -i/--images, -o/--output
```

Setelah itu masukkan perintah terminalnya

```
C:\Users\Acer 4752\Downloads\imagest>python stitch.py --images images/scottsdale --output output.png
[INFO] loading images...
[INFO] stitching images...
[INFO] Stitching Status: 0
```

Apabila tidak ada error di code maka akan muncul output gambar yang telah di stitch seperti ini