LAPORAN 5 PEMROSESAN PARALEL
EKSEKUSI PROGRAM IMAGE STITCHING

Oleh:
Muhammad Zihni Athalla
09011282227084
SK5C

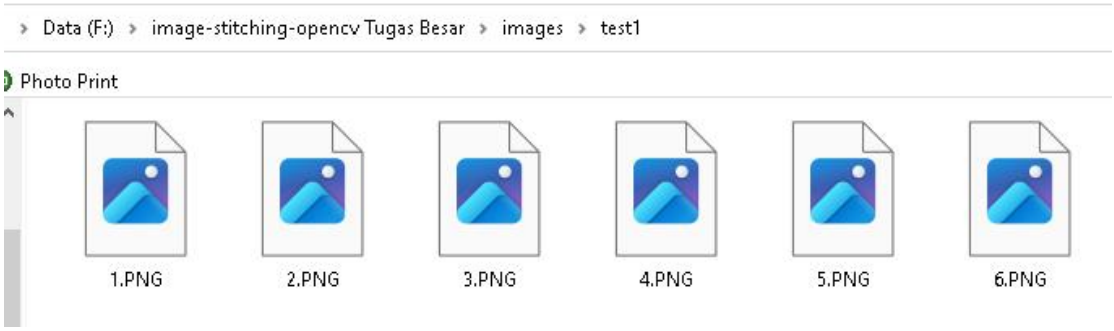
Dosen Pengampu:
Adi Hermansyah, S.Kom., M.T.

PROGRAM STUDI SISTEM KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA
PALEMBANG
2023

Alat dan Bahan

1. Sistem operasi (Windows/Linux/mac)
2. Python
3. Visual Studio Code
4. CMD

Gambar yang terpisah untuk image stitching



Diatas merupakan gambar yang terpotong dari peta google earth Fasilkom kampus indralaya

Mengecek status python pada komputer

<u>Menginstall beberapa program dan utility yang dibutuhkan untuk image stitching</u>

Program PIP

Imutils

Opencv

Numpy

Program python untuk menjalankan image stitching

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

# import the necessary packages
from imutils import paths
import numpy as np
import argparse
import imutils
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 stich list
for imagePath in imagePaths:
    image = cv2.imread(imagePath)
    images.append(image)


# initialize OpenCV's image stitcher object and then perform the image
# stitching
```

```python
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 adjustment...")

    # Manually set camera parameters
    stitcher.setWarper(cv2.detail_WaveCorrectKind_HORIZ)
    stitcher.setWaveCorrection(True)
    stitcher.setFeaturesFinder(cv2.Stitcher_createFeaturesFinder())

    # 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)

```

```python
# 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.")

# ... (existing code)
```

Eksekusi program image stitching dengan CMD

Masuk kedalam file tempat program dan gambar untuk stitching berada

```
C:\WINDOWS\system32>f:

F:\>cd image-stitching-opencv Tugas Besar

F:\image-stitching-opencv Tugas Besar>
```

Running program image stitching

```
F:\image-stitching-opencv Tugas Besar>python image_stitching_simple.py --images images/test1 --output output.png
[INFO] loading images...
[INFO] stitching images...
[INFO] Stitching Status: 0
```

Hasil stitching