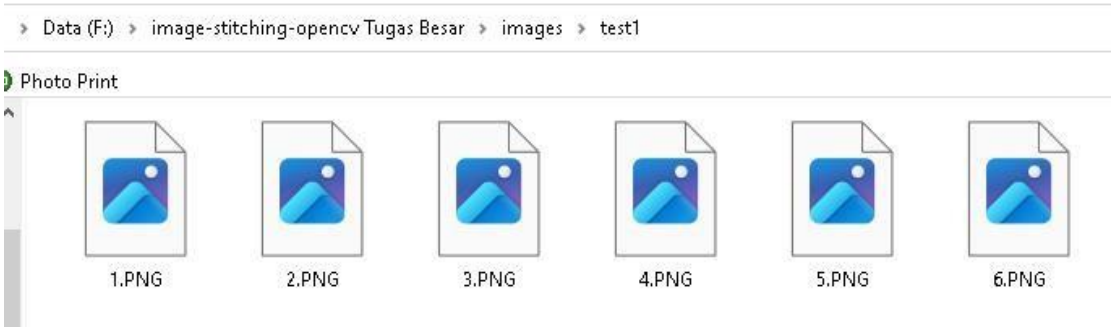
LAPORAN 5 PEMROSESAN PARALEL
EKSEKUSI PROGRAM IMAGE STITCHING

Oleh:
Pratama Arjan Rangkuti
09011182227006
SK5C

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

PROGRAM STUDI SISTEM KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA
PALEMBANG
2023

Gambar yang terpisah untuk image stitching



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

Mengecek status python pada komputer

```
C:\Users\Apiterbakar>python --version
Python 3.11.4
```

Menginstall beberapa program dan utility yang dibutuhkan untuk image stitching

Imutils

```
C:\Users\Apiterbakar>pip install imutils
Requirement already satisfied: imutils in c:\users\apiterbakar\appdata\local\programs\python\python311\lib\site-packages
 (0.5.4)
```

Opencv

```
C:\Users\Apiterbakar>pip install opencv-python
Requirement already satisfied: opencv-python in c:\users\apiterbakar\appdata\local\programs\python\python311\lib\site-pa
ckages (4.8.1.78)
```

Numpy

```
C:\Users\Apiterbakar>pip install numpy
Requirement already satisfied: numpy in c:\users\apiterbakar\appdata\local\programs\python\python311\lib\site-packages (
1.26.2)
```

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)

```

```
64  # otherwise, the stitching failed
65  else:
66      print("[INFO] image stitching failed ({})".format(status))
67
68      # print additional information
69      if status == cv2.Stitcher_ERR_NEED_MORE_IMGS:
70          print("[INFO] Need more images for stitching.")
71      elif status == cv2.Stitcher_ERR_HOMOGRAPHY_EST_FAIL:
72          print("[INFO] Homography estimation failed.")
73      elif status == cv2.Stitcher_ERR_CAMERA_PARAMS_ADJUST_FAIL:
74          print("[INFO] Camera parameters adjustment failed.")
75      elif status == cv2.Stitcher_ERR_MATCH_CONFIDENCE_FAIL:
76          print("[INFO] Match confidence test failed.")
77      elif status == cv2.Stitcher_ERR_CAMERA_PARAMS_VERIFY_FAIL:
78          print("[INFO] Camera parameters verification failed.")
79
80  # ... (existing code)
```

Hasil stitching