

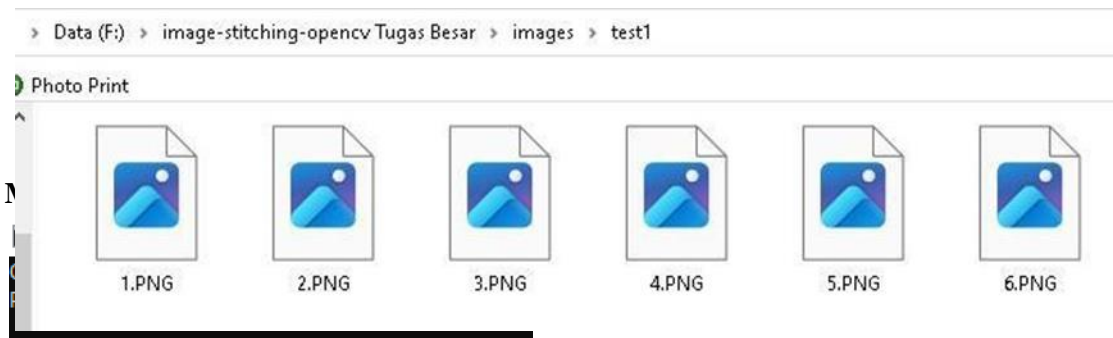
LAPORAN PEMROSESAN PARALEL
“EKSEKUSI PROGRAM IMAGE STITCHING”



Nama : Muhammad Daffa Maulana
NIM : 09011282227063
Jurusan : Sistem Komputer
Dosen : Adi Hermansyah, M.T

PROGRAM STUDI SISTEM KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA

Bahan/gambar untuk image stitching



Menginstall beberapa program yang dibutuhkan

Imutil

```
C:\Users\USER>pip install imutils
Collecting imutils
  Downloading imutils-0.5.4.tar.gz (17 kB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Building wheels for collected packages: imutils
  Building wheel for imutils (pyproject.toml) ... done
  Created wheel for imutils: filename=imutils-0.5.4-py3-none-any.whl s
  Stored in directory: c:\users\user\appdata\local\pip\cache\wheels\31
Successfully built imutils
Installing collected packages: imutils
Successfully installed imutils-0.5.4
```

Opencv

```
CA: Command Prompt
C:\Users\USER>pip install opencv-python
Collecting opencv-python
  Obtaining dependency information for opencv-python from https://files.pythonhoste
ncv_python-4.8.1.78-cp37-abi3-win_amd64.whl.metadata
  Downloading opencv_python-4.8.1.78-cp37-abi3-win_amd64.whl.metadata (20 kB)
```

Numpy

```
CA: Command Prompt
[notice] To update, run: C:\Users\USER\AppData\Local\Microsoft\WindowsApps\
C:\Users\USER>pip install numpy
Requirement already satisfied: numpy in c:\users\user\appdata\local\packages
-packages (1.26.2)
[notice] A new release of pip is available: 23.2.1 -> 23.3.1
[notice] To update, run: C:\Users\USER\AppData\Local\Microsoft\WindowsApps\
```

Program yang diperlukan untuk melakukan image stitching

```
1  # USAGE
2  # python image_stitching_simple.py --images images/scottsdale --output output.png
3
4  # import the necessary packages
5  from imutils import paths
6  import numpy as np
7  import argparse
8  import imutils
9  import cv2
10
11 # construct the argument parser and parse the arguments
12 ap = argparse.ArgumentParser()
13 ap.add_argument("-i", "--images", type=str, required=True,
14                 help="path to input directory of images to stitch")
15 ap.add_argument("-o", "--output", type=str, required=True,
16                 help="path to the output image")
17 args = vars(ap.parse_args())
18
19 # grab the paths to the input images and initialize our images list
20 print("[INFO] loading images...")
21 imagePathPaths = sorted(list(paths.list_images(args["images"])))
22 images = []
23
24 # loop over the image paths, load each one, and add them to our
25 # images to stitch list
26 for imagePath in imagePathPaths:
27     image = cv2.imread(imagePath)
28     images.append(image)
29
30
31 # initialize OpenCV's image sticher object and then perform the image
32 # stitching
```

```
33 print("[INFO] stitching images...")
34
35 # Create a Sticher with a default ORB (feature-based) detector
36 sticher = cv2.Stitcher_create(cv2.Stitcher_SCANS)
37
38 # Detect keypoints and set camera parameters manually
39 status, stitched = sticher.stitch(images)
40 if status != cv2.Stitcher_OK:
41     print("[INFO] Camera parameters adjustment failed. Retrying with manual adjustment...")
42
43     # Manually set camera parameters
44     sticher.setWarper(cv2.detail_WaveCorrectKind_HORIZ)
45     sticher.setWaveCorrection(True)
46     sticher.setFeaturesFinder(cv2.Stitcher_createFeaturesFinder())
47
48     # Retry stitching
49     status, stitched = sticher.stitch(images)
50
51 # print additional information
52 print("[INFO] Stching Status:", status)
53
54 # if the status is '0', then OpenCV successfully performed image
55 # stitching
56 if status == cv2.Stitcher_OK:
57     # write the output stitched image to disk
58     cv2.imwrite(args["output"], stitched)
59
60     # display the output stitched image to our screen
61     cv2.imshow("Stitched", stitched)
62     cv2.waitKey(0)
63
```

```

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)

```

Output

