

## **LAPORAN PRATIKUM IMAGE STITCHING**



**Disusun oleh:**

Nama : Nanda ausil Jannati

NIM : 09011282227096

Kelas : SK5C

Mata Kuliah : Pemrosesan Paralel

Dosen Pengampu : Adi Hermansyah, M. T.

**PROGRAM STUDI SISTEM KOMPUTER  
FAKULTAS ILMU KOMPUTER UNIVERSITAS  
SRIWIJAYA  
2023**

## 1.1 Proses-proses Dalam Image Stitching

Proses image stitching bisa dilihat dalam proses dibawah ini:

1. Input image semua potongan image yang akan dijadikan satu foto diinput ke program atau software.
2. Features Extraction gambar yang telah diinput sebelumnya akan ditandai seperti titik-titik. Bagian yang ditandai biasanya bersifat unik dan memiliki kesamaan dengan gambar setelah atau keseluruhan gambar.
3. Feature Matching  
Pada area yang telah ditandai akan dibandingkan bagian yang sama. Hal ini bertujuan untuk menemukan tempat gambar yang bisa di “tumpang tindih” dari titik tersebut.
4. Model estimation & image warping proses ini akan memetakan foto-foto yang berhubungan sesuai satu sama lain.
5. Image blending setelah pemetaan dan posisi foto yang ingin digabung telah selesai, langkah terakhir adalah penggabungan foto tersebut hingga menjadi foto panorama hasil penggabungan semua input tadi.
6. Output image hasilnya akan keluar sebagai foto panorama.

## 1.2 Instal Packages Yang Akan Digunakan

Tujuan dari praktikum ini adalah untuk mengetahui cara kerja dari image stitching. Beberapa software telah disiapkan untuk praktikum image stitching seperti VM (Virtual Machine) tipe GUI yaitu ubuntu Desktop lalu install beberapa library python sebagai berikut:

1. Menginstal pip3 install imutils

Untuk menginstal library gunakan perintah di bawah ini. di sini sudah terinstal pada ubuntu desktop

```
nanda@nanda-VirtualBox:~$ pip3 install imutils
Defaulting to user installation because normal site-packages is not writeable
Collecting imutils
  Downloading imutils-0.5.4.tar.gz (17 kB)
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: imutils
  Building wheel for imutils (setup.py) ... done
  Created wheel for imutils: filename=imutils-0.5.4-py3-none-any.whl size=25858 sha256=12490d795b8bc12f3c1c21cce44991d1f0dd774416228520f8feb848019c7871
  Stored in directory: /home/nanda/.cache/pip/wheels/85/cf/3a/e265e975a1e7c7e54eb3692d6aa4e2e7d6a3945d29da46f2d7
Successfully built imutils
Installing collected packages: imutils
Successfully installed imutils-0.5.4
```

*Gambar 2. 1 pip3 install imutils*

## 2. Menginstal pip3 install opencv-python

```
nanda@nanda-VirtualBox:~$ pip3 install opencv-python
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: opencv-python in ./local/lib/python3.10/site-packages (4.8.1.78)
Requirement already satisfied: numpy>=1.19.3 in ./local/lib/python3.10/site-packages (from opencv-python) (1.26.2)
```

*Gambar 2. 2 pip3 install opencv-python*

## 3. Menginstal pip3 install matplotlib

```
nanda@nanda-VirtualBox:~$ pip3 install matplotlib
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: matplotlib in ./local/lib/python3.10/site-packages (3.8.2)
Requirement already satisfied: packaging>=20.0 in /usr/lib/python3/dist-packages (from matplotlib) (21.3)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/lib/python3/dist-packages (from matplotlib) (2.4.7)
Requirement already satisfied: fonttools>=4.22.0 in ./local/lib/python3.10/site-packages (from matplotlib) (4.44.3)
Requirement already satisfied: contourpy>=1.0.1 in ./local/lib/python3.10/site-packages (from matplotlib) (1.2.0)
Requirement already satisfied: python-dateutil>=2.7 in ./local/lib/python3.10/site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: pillow>=8 in /usr/lib/python3/dist-packages (from matplotlib) (9.0.1)
Requirement already satisfied: numpy<2,>=1.21 in ./local/lib/python3.10/site-packages (from matplotlib) (1.26.2)
Requirement already satisfied: cycler>=0.10 in ./local/lib/python3.10/site-packages (from matplotlib) (0.12.1)
Requirement already satisfied: kiwisolver>=1.3.1 in ./local/lib/python3.10/site-packages (from matplotlib) (1.4.5)
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
```

*Gambar 2. 3 Menginstal pip3 install matplotlib*

## 1.3 Eksekusi terhadap image stitching

1. Masuk ke dalam direktori yang berisi codingan image stitching dan di dalamnya terdapat direktori image yang akan di gabungkan

```
ubuntu@ubuntu-desktop:~/Downloads/Image-stitching-opencv Tugas Besar-20231118T101554Z-001/Image-stitching-opencv Tugas Besar$ ll
total 4068
drwxrwxr-x 3 ubuntu ubuntu 4096 Nov 18 17:30 ./
drwxrwxr-x 3 ubuntu ubuntu 4096 Nov 18 17:22 ../
drwxrwxr-x 3 ubuntu ubuntu 4096 Nov 18 17:22 images/
-rw-rw-r-- 1 ubuntu ubuntu 1552 Nov 16 16:40 image_stitching_simple.py
-rw-rw-r-- 1 ubuntu ubuntu 4144665 Nov 18 17:30 output.png
-rw-rw-r-- 1 ubuntu ubuntu 166 Nov 16 16:40 'perintah terminal'
```

*Gambar 3. 1 File dari image stitching*

## 2. Image



*Gambar 3. 2 Panorama gunung*

### 3. codingan python

```
ubuntu@ubuntu-desktop: /Downloads/image-stitching-opencv tugas Besar 202211181015542-901/image-stitching-opencv tugas Besar$ cat image_stitching_simple.py
# 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 stitch list
for imagePath in imagePaths:
    image = cv2.imread(imagePath)
    images.append(image)

# initialize OpenCV's image sticher object and then perform the image
# stitching
print("[INFO] stitching images...")
stitcher = cv2.createStitcher() if imutils.is_cv3() else cv2.Stitcher_create()
(status, stitched) = stitcher.stitch(images)

# if the status is '0', then OpenCV successfully performed image
# stitching
if status == 0:
    # 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, likely due to not enough keypoints
# being detected
```

*Gambar 3. 3 Code python image stitching*

### 4. Menjalankan code python

Kemudian lakukan perintah di terminal ubuntu dektop dengan perintah sebagai berikut.

```
python image_stitching_simple.py --images images/scottsdale --output output.png  
python image_stitching.py --images images/scottsdale --output output.jpg \ --crop 1
```

*Gambar 3. 4 Perintah untuk menjalankan image stitching*

## 5. Output



*Gambar 3. 5 Hasil output dari image stitching*