# LAPORAN PEMROSESAN PARALEL (IMAGE STITCHING DENGAN PYTHON TANPA MPI)



Nama : Khairunnisa Junaidi

NIM : 09011282227072

Kelas : SK3C

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

# FAKULTAS ILMU KOMPUTER

**PROGRAM STUDI SISTEM KOMPUTER UNIVERSITAS SRIWIJAYA**

# 2023

## PENJELASAN SINGKAT IMAGE STITCHING

## Image stitching adalah proses menggabungkan beberapa gambar fotografi dengan bidang pandang yang tumpang tindih untuk menghasilkan panorama tersegmentasi atau gambar resolusi tinggi. Umumnya dilakukan melalui penggunaan perangkat lunak komputer, dan kebanyakan pendekatan untuk image stitching memerlukan tumpang tindih yang hampir sama persis di antara gambar dan eksposur yang identik untuk menghasilkan hasil yang mulus. Proses ini dapat dibagi menjadi tiga komponen utama: registrasi gambar, kalibrasi, dan blending. Registrasi gambar melibatkan pencocokan fitur antara gambar, sedangkan kalibrasi melibatkan penyesuaian parameter kamera. Tahap terakhir, blending, melibatkan penggabungan gambar yang telah diregistrasi dan dikalibrasi menjadi satu gambar yang mulus.

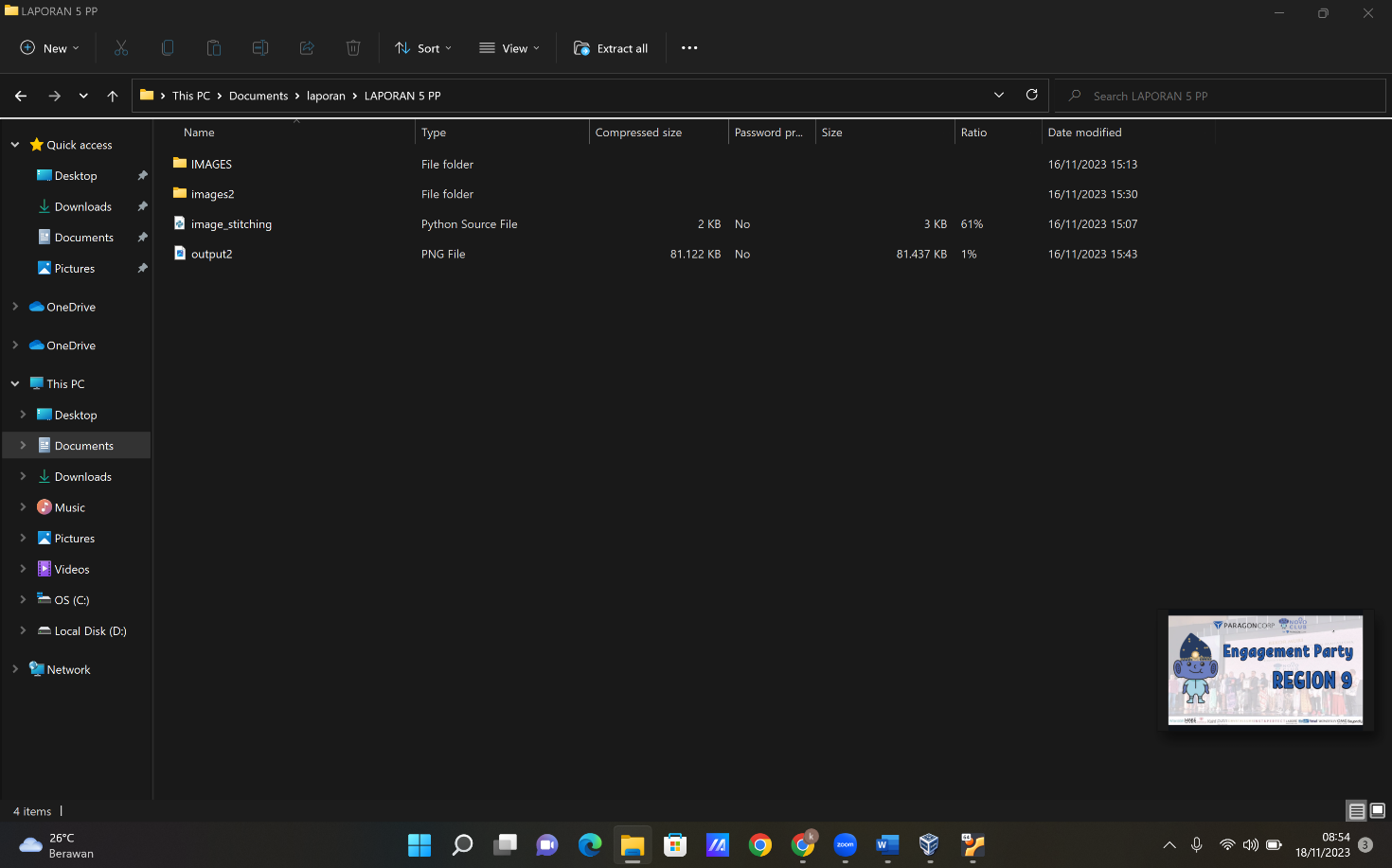
## Image stitching telah menjadi revolusi dalam dunia fotografi dan digunakan dalam berbagai aplikasi, seperti pembuatan panorama, stabilisasi gambar, pemetaan digital, citra medis, dan lainnya. Proses image stitching juga dapat melibatkan deteksi fitur, pencocokan fitur, dan penggunaan algoritma seperti RANSAC (RANdom SAmple Consensus) untuk menemukan transformasi yang akurat antara gambar-gambar yang akan digabungkan.

## Dalam konteks aplikasi praktis, image stitching dapat dilakukan menggunakan perangkat lunak seperti PTGui atau dengan menerapkan algoritma image stitching menggunakan library seperti OpenCV pada Python. Dengan demikian, image stitching memungkinkan penggabungan gambar-gambar menjadi panorama yang mulus dan gambar resolusi tinggi.

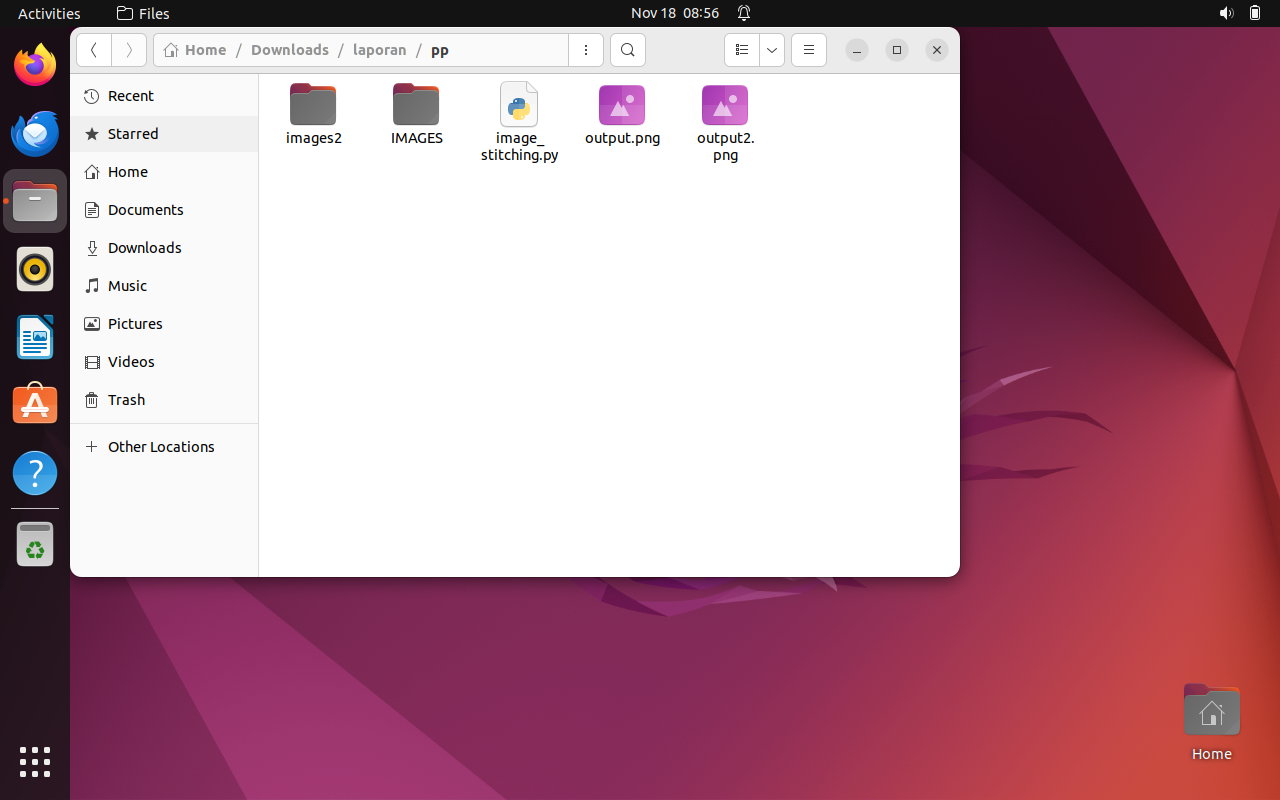
## LANGKAH-LANGKAH IMPLEMENTASI IMAGE STITCHING TANPA MPI

1. Buatlah sebuah folder yang berisikan potongan-potongan gambar, kodingan python dan juga output gambar. Lakukan step ini pada windows dan ubuntu desktop.

Windows:



Ubuntu Desktop:



## Gambar yang digunakan:

Dapat di ambil melalui Google Earth



Codingan yang digunakan:

# 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

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)

# 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.")

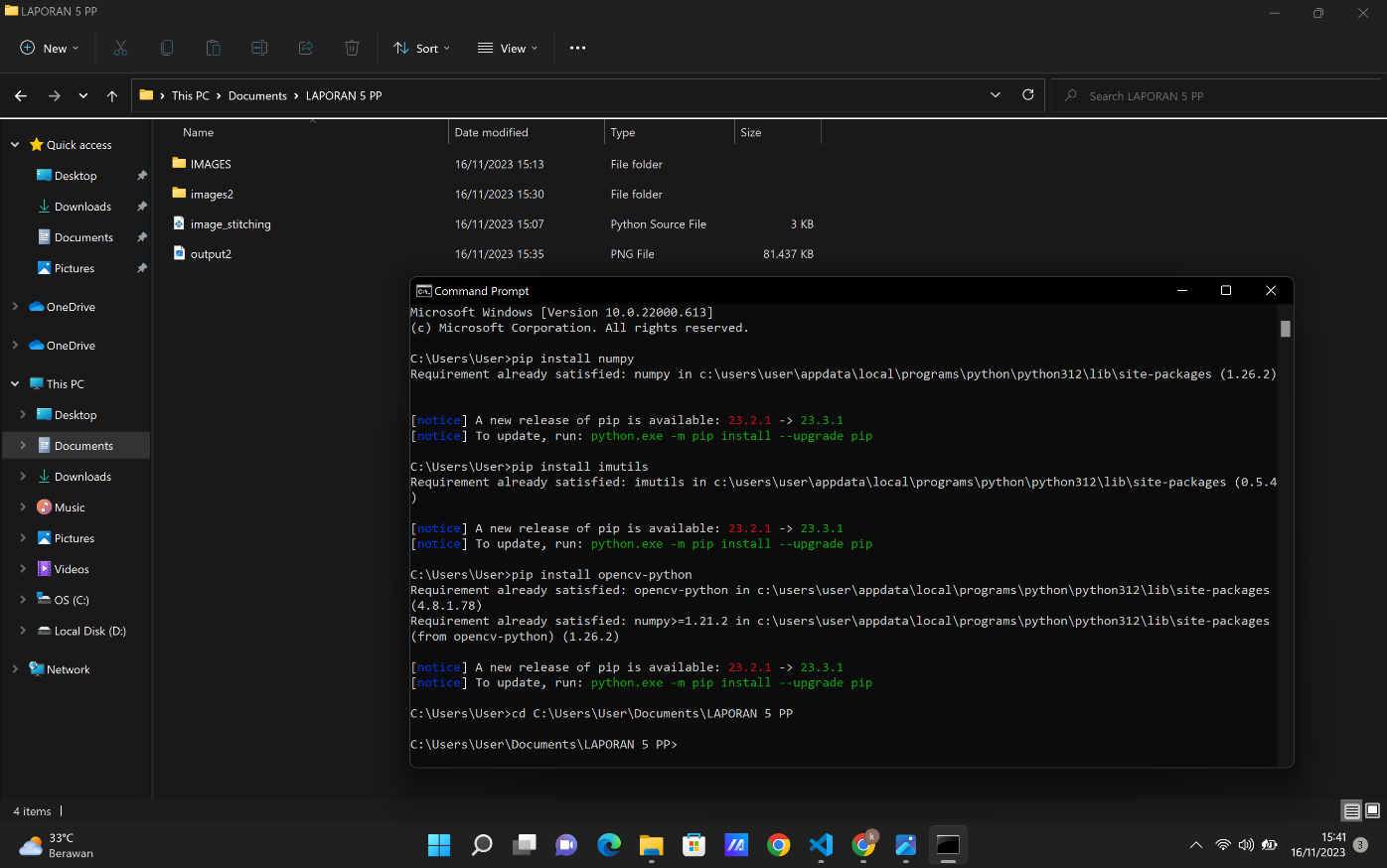
elif status == cv2.Stitcher\_ERR\_CAMERA\_PARAMS\_ADJUST\_FAIL: print("[INFO] Camera parameters adjustment failed.")

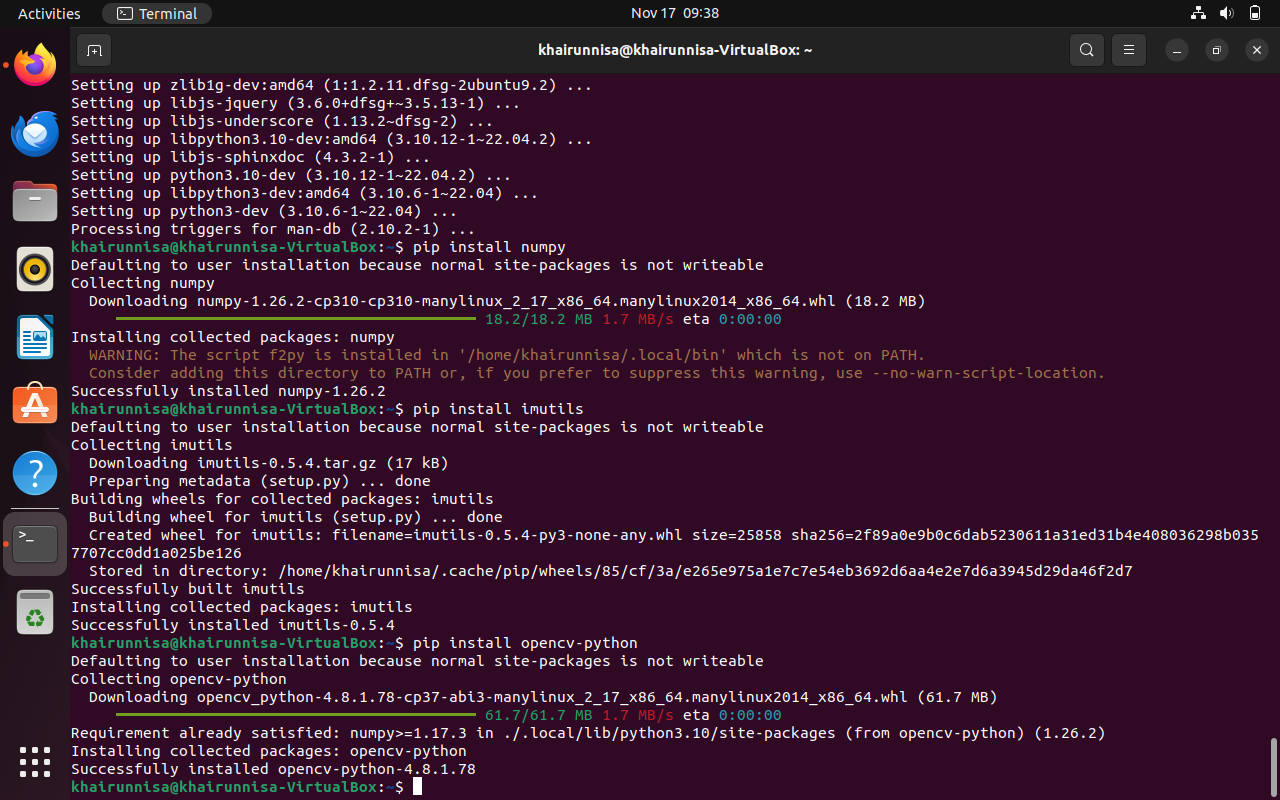
elif status == cv2.Stitcher\_ERR\_MATCH\_CONFIDENCE\_FAIL: print("[INFO] Match confidence test failed.")

elif status == cv2.Stitcher\_ERR\_CAMERA\_PARAMS\_VERIFY\_FAIL: print("[INFO] Camera parameters verification failed.")

# ... (existing code)

1. Kemudian, install paket image stitching seperti imutils, numpy dan opencv-python dengan command ”pip install imutils”, “pip install numpy”, dan “pip install opencv-python” pada cmd dan juga ubuntu desktop.





Tujuan menginstal ketiga library ini adalah untuk mempermudah pengolahan citra dan komputasi ilmiah pada Python. Dalam konteks image stitching, OpenCV dan imutils digunakan untuk menggabungkan beberapa gambar menjadi satu gambar panorama dengan resolusi tinggi. Sedangkan Numpy digunakan untuk melakukan operasi matematika pada array dan matriks, yang sering digunakan dalam pengolahan citra. Dengan menginstal ketiga library ini, penggunaan python dapat dengan mudah melakukan pengolahan citra dan komputasi ilmiah pada Python.

1. Menjalankan paket image stitching di ubuntu desktop dan cmd menggunakan skrip python, dan setelah di running akan menghasilkan:

