LAPORAN PEMROSESAN PARALEL
(IMAGE STITCHING DENGAN PYTHON TANPA MPI)

Nama : Eka Ratna Anindita

NIM : 09011282227045

Kelas : SK5C

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

FAKULTAS ILMU KOMPUTER

PROGRAM STUDI SISTEM KOMPUTER

UNIVERSITAS SRIWIJAYA

2023

- **PENJELASAN SINGKAT IMAGE STITCHING**

Image stitching adalah metode untuk menggabungkan sejumlah gambar menjadi satu gambar panoramik yang lebih besar. Tujuannya adalah menciptakan representasi visual yang komprehensif dari suatu adegan atau objek yang sulit dimuat dalam satu gambar. Proses ini melibatkan beberapa langkah, termasuk identifikasi fitur pada setiap gambar, penyesuaian perspektif, dan penggabungan gambar untuk membentuk panorama. Proses image stitching umumnya melibatkan:

- Identifikasi Fitur: Mengenali dan mengekstraksi fitur khas pada masing-masing gambar sebagai titik referensi untuk penyesuaian perspektif.
- Penyesuaian Perspektif: Memastikan bahwa semua gambar memiliki perspektif dan orientasi yang serupa melalui transformasi geometris, seperti pergeseran, rotasi, dan skalasi.
- Penggabungan Gambar: Menyatukan gambar yang telah disesuaikan menjadi satu gambar panoramik dengan menggabungkan piksel dari area tumpang tindih.
- Penanganan Tantangan: Mengatasi masalah seperti pergeseran perspektif, perbedaan intensitas cahaya, dan distorsi lensa yang mungkin muncul selama pengambilan gambar.

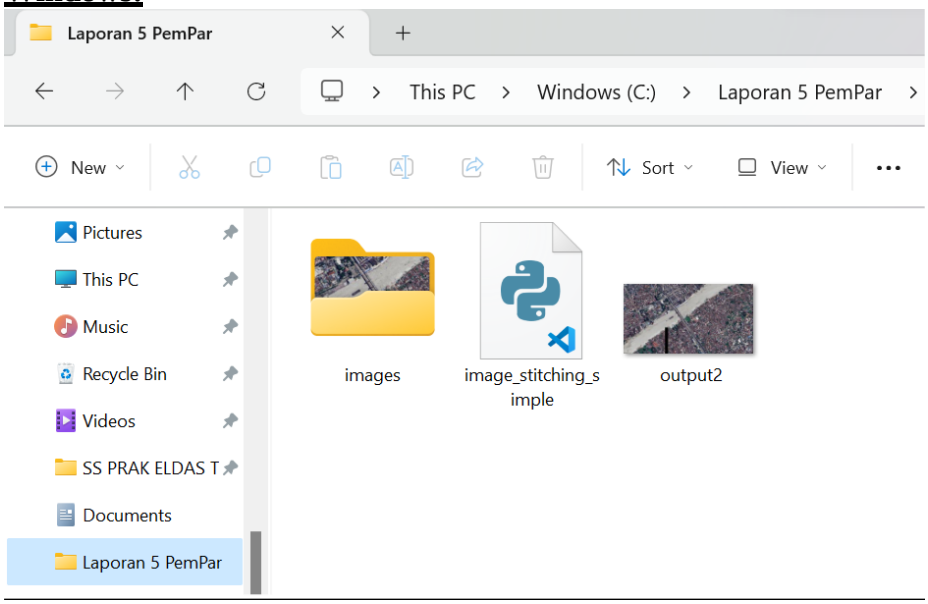
Hasil akhir dari proses image stitching adalah gambar panorama yang memberikan tampilan komprehensif dari adegan yang direkam. Teknik ini banyak digunakan dalam fotografi lanskap, fotografi arsitektur, serta aplikasi realitas virtual atau augmented. Dengan menggunakan Python dan pustaka seperti OpenCV, implementasi image stitching dapat dilakukan dengan efisien.

- **LANGKAH-LANGKAH IMPLEMENTASI IMAGE STITCHING TANPA MPI**

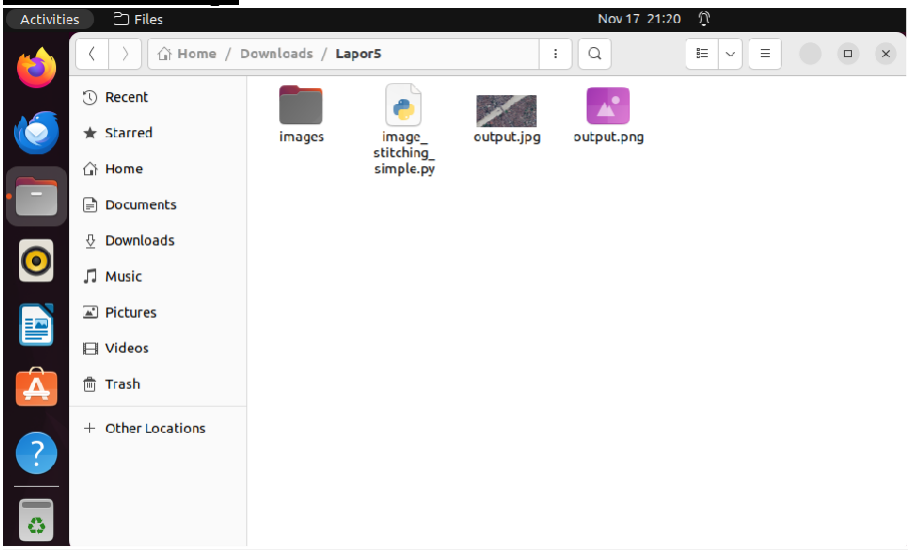
SEBELUM BEKERJA

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

Windows:

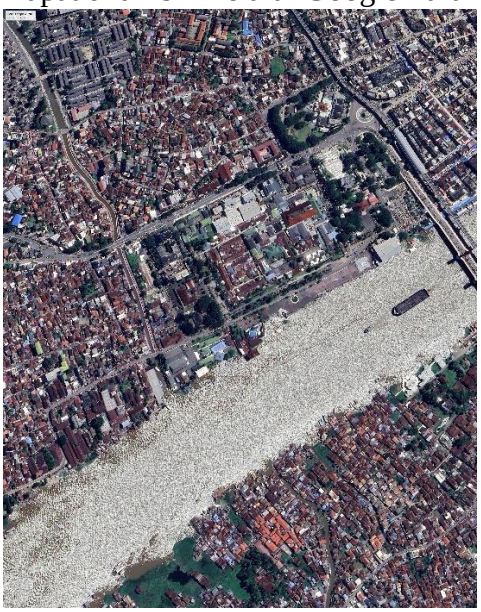


Ubuntu Desktop:



Gambar yang digunakan:

Dapat di ambil melalui Google Earth



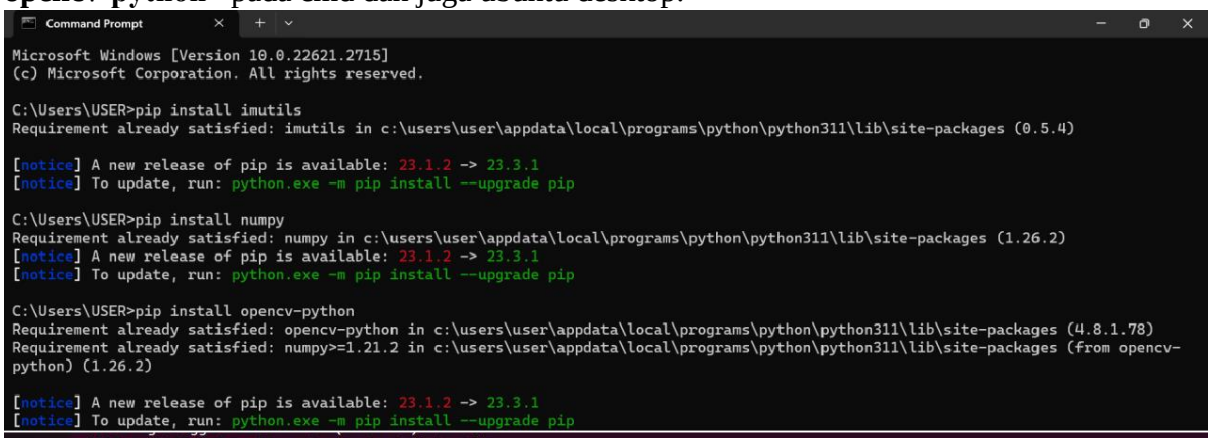
Output:

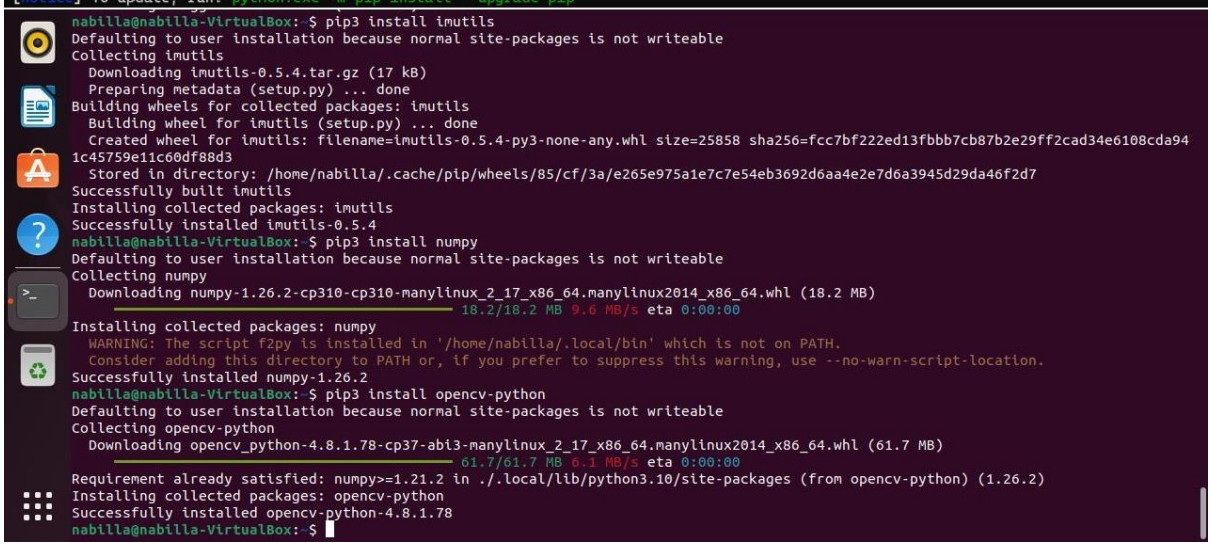
Program atau Kodingan yang Saya Gunakan:

```
# 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)
```

2. Kemudian, instalasi pustaka-putsaka 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.

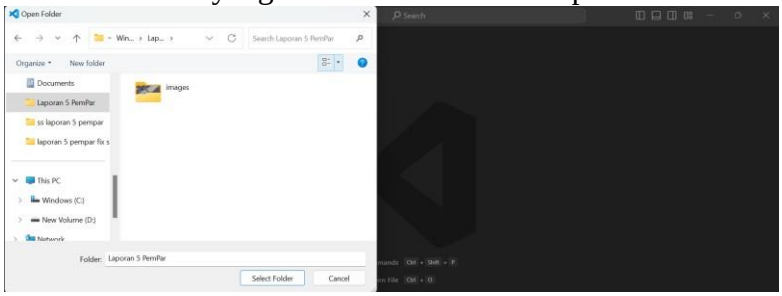


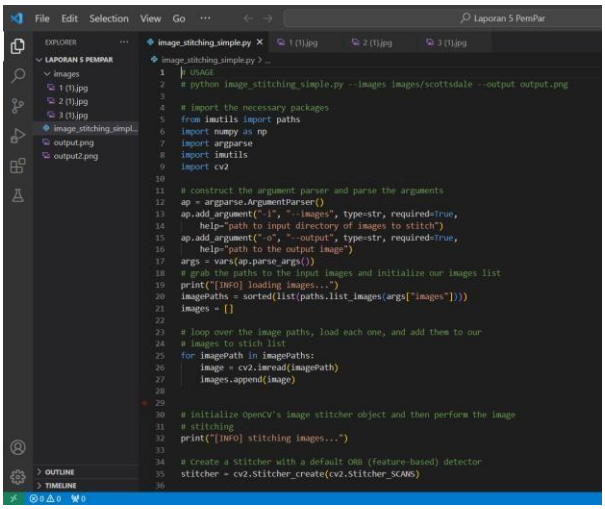


-Tujuan menginstal imutils adalah untuk mempermudah pengembangan aplikasi pengolahan gambar dan video dengan menggunakan OpenCV.
-Instalasi NumPy pada proyek image stitching memiliki beberapa tujuan utama, karena NumPy menyediakan sejumlah fitur dan fungsionalitas yang sangat bermanfaat dalam konteks pengolahan gambar dan manipulasi array.
-opencv-python (atau OpenCV) adalah pustaka populer untuk pengolahan gambar dan komputer vision. Menyediakan algoritma-algoritma pengolahan gambar, operasi pemrosesan citra, deteksi objek, dan fungsi-fungsi lainnya yang diperlukan dalam aplikasi pengolahan gambar.

A. Image Stitching Pada VSCode

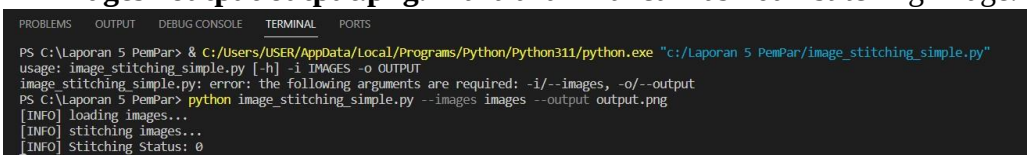
1. Buka folder yang telah kita buat di awal pada VSCode





```
# 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)
```

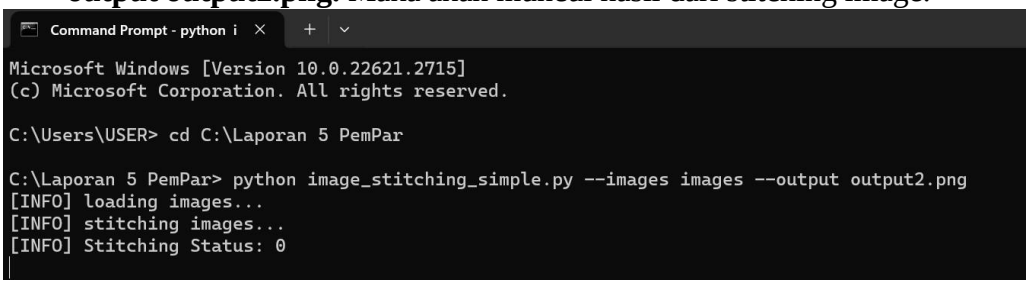
2. Kemudian jalankan kodingan tersebut dan akan muncul output pada terminal seperti di bawah, lalu ketikan di **terminal python image_stitching_simple.py –images images –output output.png.** Maka akan muncul hasil dari stitching image.



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Laporan 5 PemPar> & C:/Users/USER/AppData/Local/Programs/Python/Python311/python.exe "c:/Laporan 5 PemPar/image_stitching_simple.py"
usage: image_stitching_simple.py [-h] -i IMAGES -o OUTPUT
image_stitching_simple.py: error: the following arguments are required: -i/--images, -o/--output
PS C:\Laporan 5 PemPar> python image_stitching_simple.py --images images --output output.png
[INFO] loading images...
[INFO] stitching images...
[INFO] Stitching Status: 0
```

B. Image Stitching pada CMD

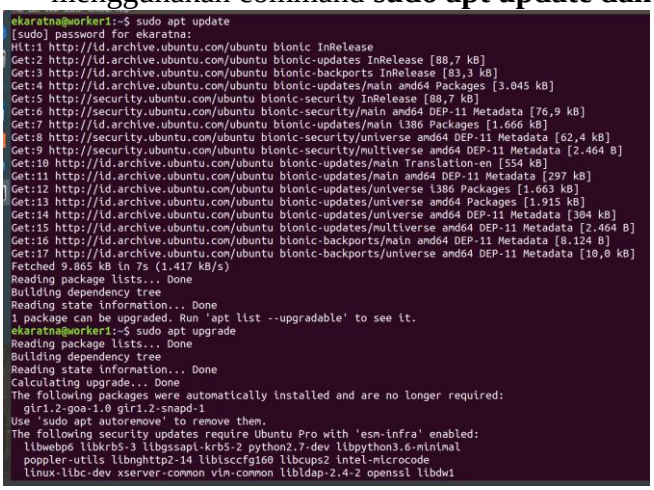
1. Masuk ke direktori dengan command **cd C:\Laporan 5 PemPar (tempat lokasi folder disimpan dan juga nama folder yang telah di buat di awal).**
2. Kemudian, ketik **command python image_stitching_simple.py –images images –output output2.png.** Maka akan muncul hasil dari stitching image.



```
Command Prompt - python i

Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. All rights reserved.

C:\Users\USER> cd C:\Laporan 5 PemPar

C:\Laporan 5 PemPar> python image_stitching_simple.py --images images --output output2.png
[INFO] loading images...
[INFO] stitching images...
[INFO] Stitching Status: 0
```

C. Image Stitching pada Ubuntu Desktop

1. Pastikan Ubuntu kalian sudah ter update dan upgrade versi terbaru, bisa menggunakan command **sudo apt update dan sudo apt upgrade.**



```
ekaratna@worker1:~$ sudo apt update
[sudo] password for ekaratna:
Hit:1 http://id.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://id.archive.ubuntu.com/ubuntu bionic-updates InRelease [88,7 kB]
Get:3 http://id.archive.ubuntu.com/ubuntu bionic-backports InRelease [83,3 kB]
Get:4 http://id.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [3.045 kB]
Get:5 http://security.ubuntu.com/ubuntu bionic-security InRelease [88,7 kB]
Get:6 http://security.ubuntu.com/ubuntu bionic-security/main amd64 DEP-11 Metadata [76,9 kB]
Get:7 http://id.archive.ubuntu.com/ubuntu bionic-updates/main i386 Packages [1.666 kB]
Get:8 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 DEP-11 Metadata [62,4 kB]
Get:9 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 DEP-11 Metadata [2.464 B]
Get:10 http://id.archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [554 kB]
Get:11 http://id.archive.ubuntu.com/ubuntu bionic-updates/main amd64 DEP-11 Metadata [297 kB]
Get:12 http://id.archive.ubuntu.com/ubuntu bionic-updates/universe i386 Packages [1.663 kB]
Get:13 http://id.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [1.915 kB]
Get:14 http://id.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 DEP-11 Metadata [304 kB]
Get:15 http://id.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 DEP-11 Metadata [2.464 B]
Get:16 http://id.archive.ubuntu.com/ubuntu bionic-backports/main amd64 DEP-11 Metadata [8.124 B]
Get:17 http://id.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 DEP-11 Metadata [10,0 kB]
Fetched 9.865 kB in 7s (1.417 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
ekaratna@worker1:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  gir1.2-goa-1.0 gir1.2-snapd-1
Use 'sudo apt autoremove' to remove them.
The following security updates require Ubuntu Pro with 'esm-infra' enabled:
  libwebp6 libkrb5-3 libgssapi-krb5-2 python2.7-dev libpython3.6-minimal
  poppler-utils libnghttp2-14 libisccfg160 libcups2 intel-microcode
  linux-libc-dev xserver-common vim-common libldap-2.4-2 openssl libdw1
```

2. Install Python3 dengan command **sudo apt install python3-pip**

```
ekaratna@worker1:~$ sudo apt install python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-pip is already the newest version (9.0.1-2.3~ubuntu1.18.04.8).
The following packages were automatically installed and are no longer required:
  gir1.2-goa-1.0 gir1.2-snapd-1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

3. Masuk ke direktori dengan command **Cd Downloads\Lapor5 (tempat penyimpanan folder dan nama folder yang kita buat di awal).**
4. Setelah masuk ke direktori ketik command **python image_stitching_simple.py –images images –output output3.png.** Maka akan muncul hasil dari stitching image.