

TUGAS KELOMPOK 11
PEMROSESAN PARALEL



DOSEN PENGAMPU :

AHMAD HERYANTO, S.KOM, M.T.
ADI HERMANSYAH, S.KOM., M.T.

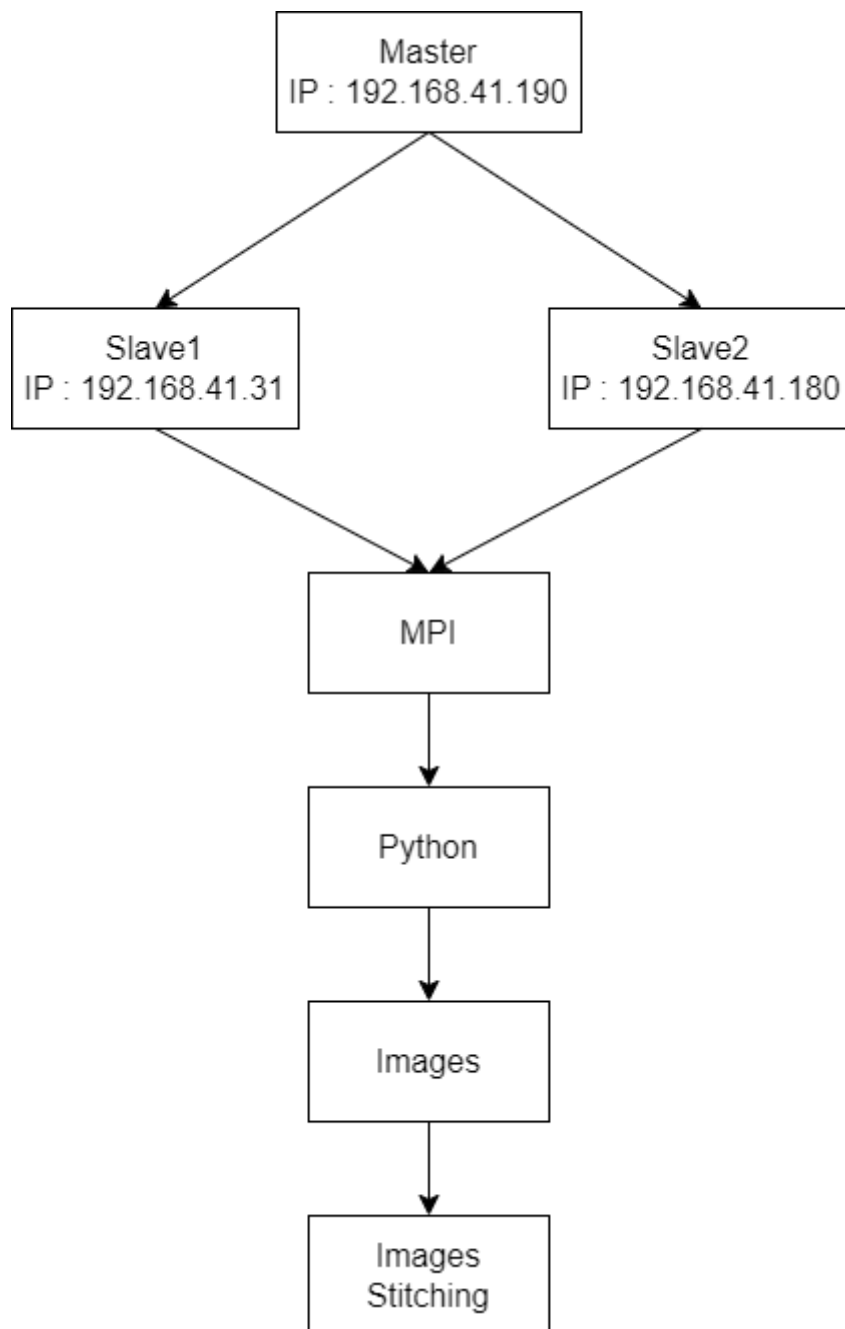
PEMBUAT (KELOMPOK 11):

MUHAMMAD RAFI RIZQULLAH	(09011282126091)
M. IKHSAN SETIAWAN	(09011282126103)
REZA PALEPI	(09011282126120)

FAKULTAS ILMU KOMPUTER
JURUSAN SISTEM KOMPUTER
UNIVERSITAS SRIWIJAYA

Laporan 5 Image Stitching

Topologi



1. Install MPI dan Python

```
sudo apt install python3-pip  
pip install mpi4py
```

2. Konfigurasi

- Konfigurasi hosts

Dapat dicek alamat ip dengan menggunakan perintah ``ip a`` ``if config`` ``hostname - I``

MASTER

192.168.41.190 master
192.168.41.31 slave1
192.168.41.180 slave2

SLAVE1

192.168.41.190 master
192.168.41.31 slave1

SLAVE2

192.168.41.190 master
192.168.41.180 slave2

- Konfigurasi SSH

Ini dapat dilakukan diberbagai hosts, seperti master dapat melakukannya pada slave1 dan slave2, sedangkan slave1 dan slave2 dapat mengkonfigurasi master. Ini dilakukan agar dapat mengetahui apakah ssh multinode telah berjalan dengan baik atau belum.

ssh <nama user>@<hosts>

3. Install module yang diperlukan

- Numpy
pip install numpy
- Imutils
pip install imutils
- Opencv
pip install opencv-python

4. Copy file pada slave

```
scp kelompokpp/* mpiuser@slave1:/home/uaskelpp/  
scp kelompokpp/* mpiuser@slave2:/home/uaskelpp/
```

5. Proses penjalanan program

- Program
USAGE
python image_stitching_simple.py --images images/scottsdale --output output.p>
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)

```

- Gambar yang digunakan
Foto1 :



Foto2 :



Foto3 :



- Running mpi4python

```
mpiexec -n 3 -host master,slave1,slave2 python3  
/home/uaskelpp/kelompokpp/ImageStitching.py -i  
/home/uaskelpp/kelompokpp/images -o OutputImageStitching.png
```

```
uaskelpp@master:~/kelompokpp$ mpiexec -n 3 -host master,slave1,slave2 python3 /h  
ome/uaskelpp/kelompokpp/ImageStitching.py -i /home/uaskelpp/kelompokpp/images -o  
OutputImageStitching.png  
[INFO] loading images...  
[INFO] loading images...  
[INFO] stitching images...  
[INFO] loading images...  
[INFO] stitching images...  
[INFO] stitching images...  
uaskelpp@master:~/kelompokpp$
```

- Output

