Laporan Pemrosesan Paralel
"Message Passing Interface pada Ubuntu Dekstop menggunakan Bahasa Python"

Nama : **Andrian Kaspari**

NIM : **09011182126038**

Jurusan : **Sistem Komputer**

Dosen : **Ahmad Heryanto, S.Kom, M.T.**

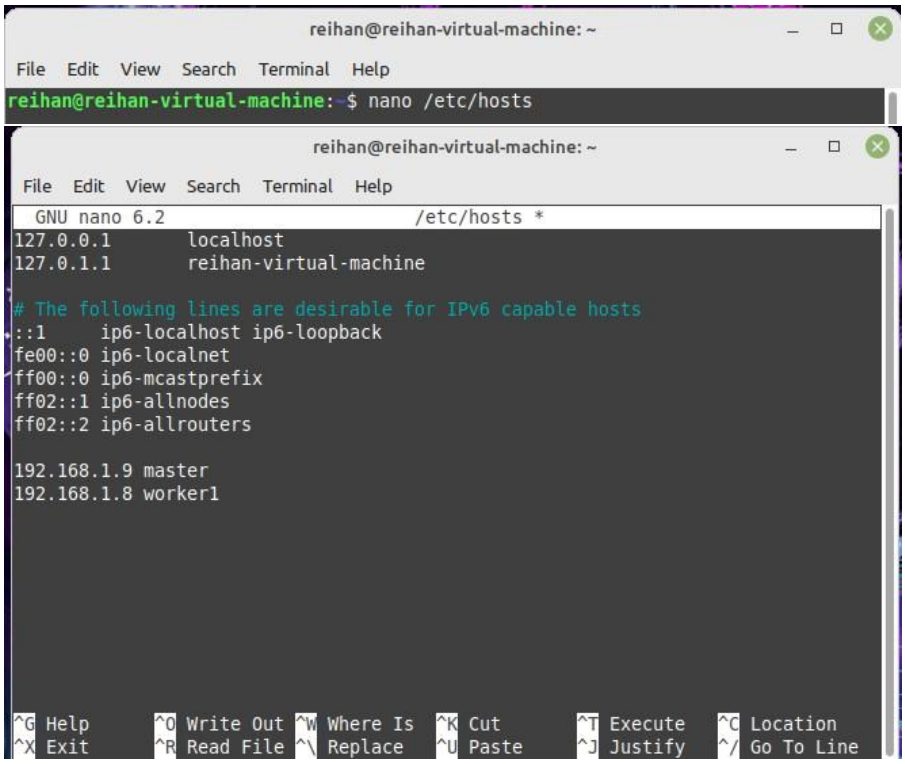
Adi Hermansyah, S.Kom., M.T.

Jurusan Sistem Komputer
Fakultas Ilmu Komputer
Universitas Sriwijaya

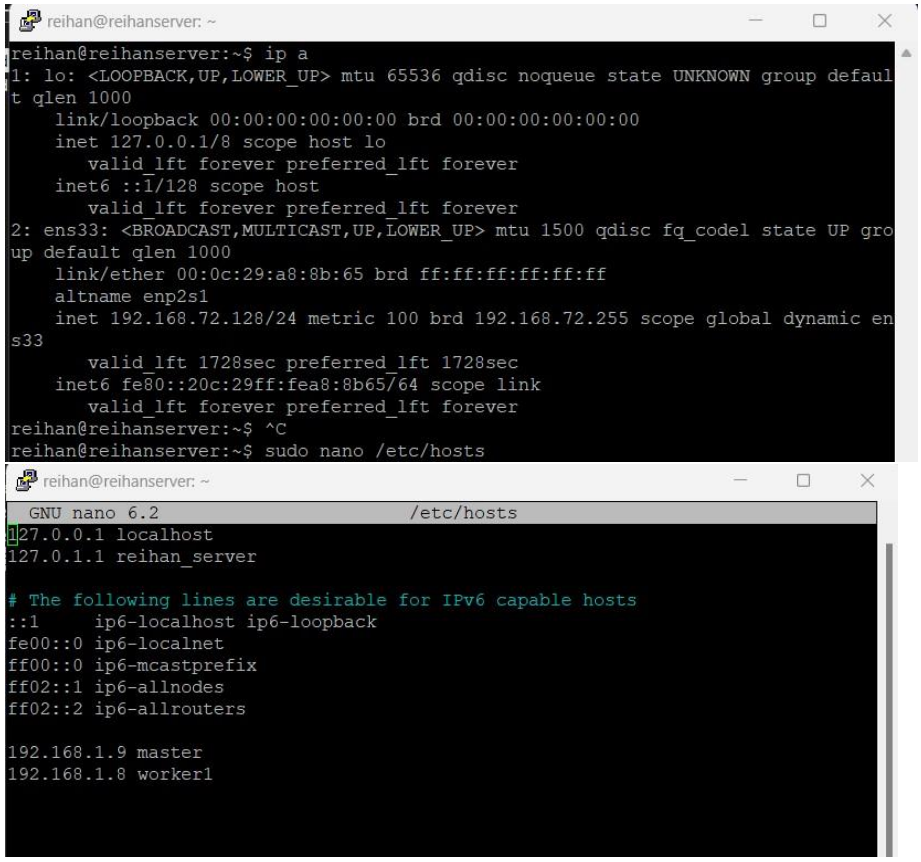
Message Passing Interface pada ubuntu dekstop menggunakan bahasa python

1. Konfigurasi file /etc/hosts

   Server

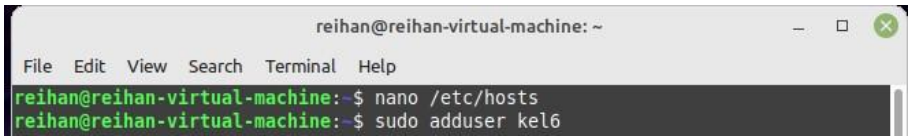


   Client

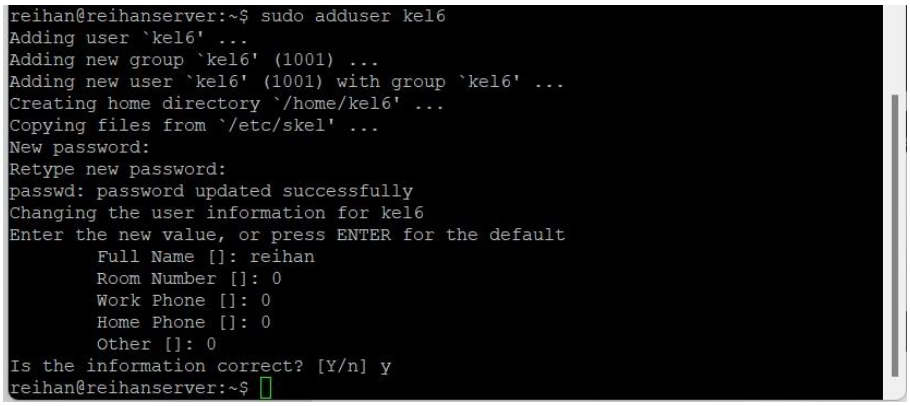


2. Menambahkan user

   Server

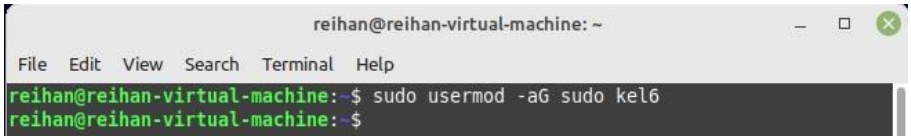


   Client

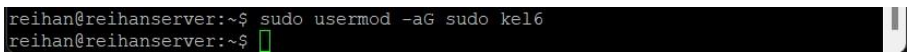


3. Memberikan akses root kepada user

   Server

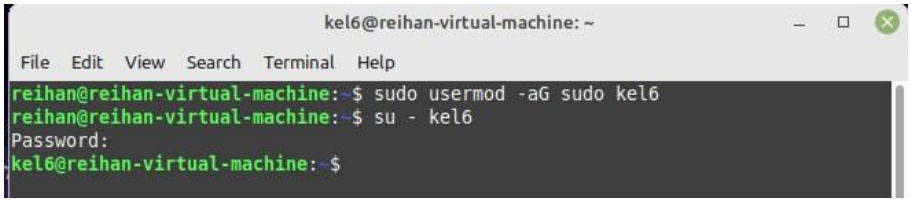


   Client

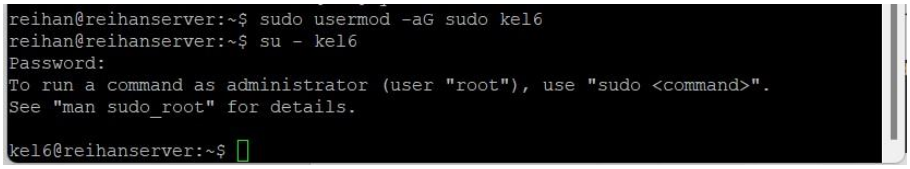


4. Melakukan login akun user

   Server

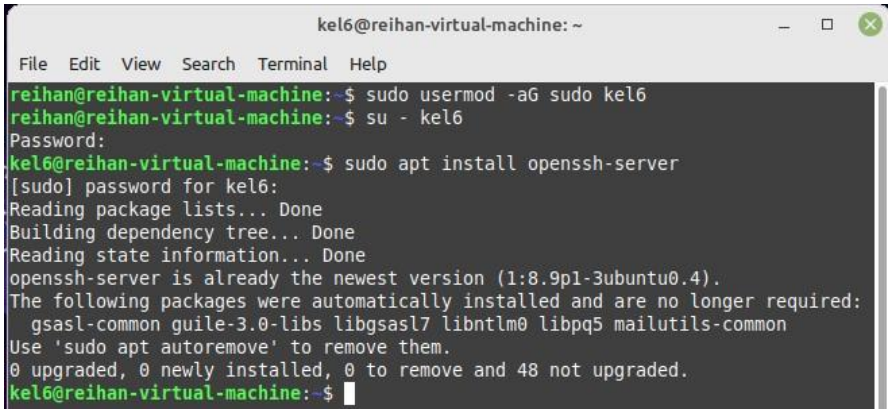


   Client

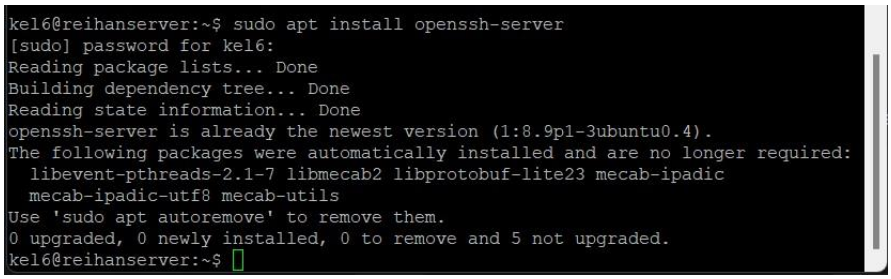


5. Instalasi paket openssh-server

   Server

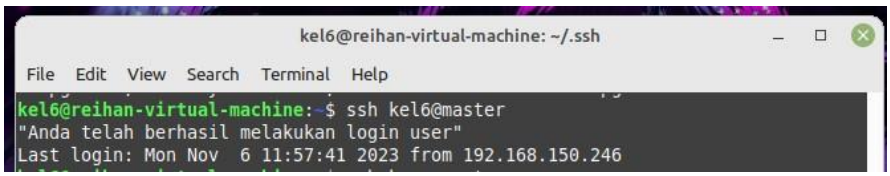


Client

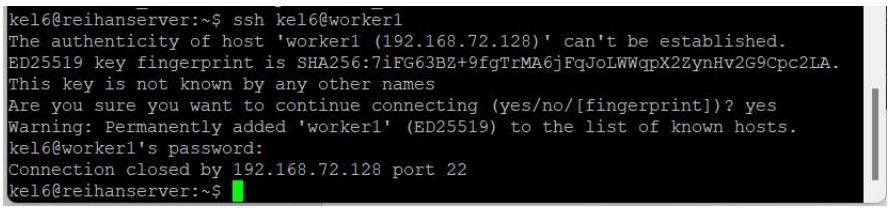


6. Pengecekan ssh

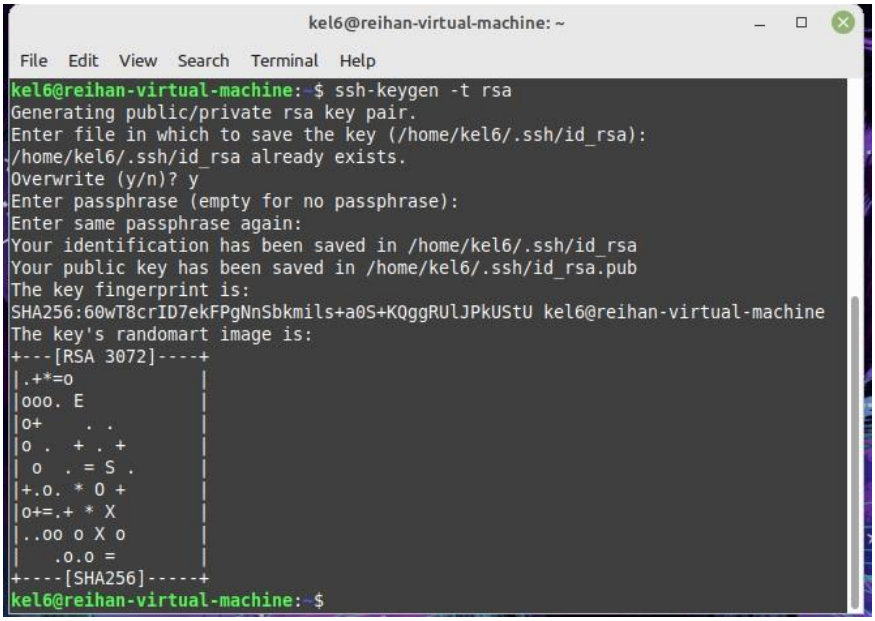
Server



Client

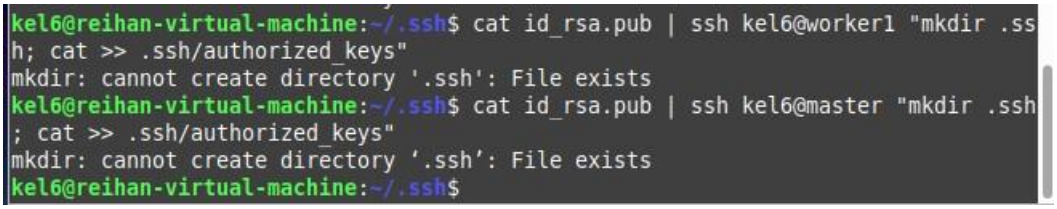


7. Membuat keygen Server



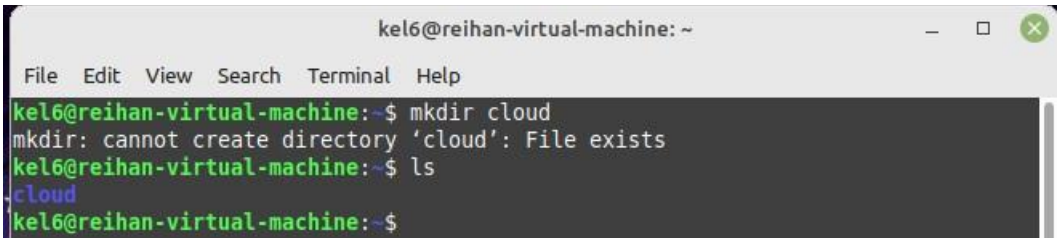
8. Menyalin keygen ke Server dan Client

   Server



9. Membuat sharing file

   Server



   Client

10. Instalasi nfs server

    Server

```
kel6@reihan-virtual-machine:~$ sudo apt install nfs-kernel-server
[sudo] password for kel6:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nfs-kernel-server is already the newest version (1:2.6.1-1ubuntu1.2).
The following packages were automatically installed and are no longer required:
  gsasl-common guile-3.0-libs libgsasl7 libntlm0 libpq5 mailutils-common
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 48 not upgraded.
kel6@reihan-virtual-machine:~$
```

11. Konfigurasi file /etc/exports

    Server

```
kel6@reihan-virtual-machine: ~
File  Edit  View  Search  Terminal  Help
kel6@reihan-virtual-machine:~$ sudo nano /etc/exports
```

```
kel6@reihan-virtual-machine: ~
File  Edit  View  Search  Terminal  Help
  GNU nano 6.2                    /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes       hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_sub>
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#

/home/kel6/cloud *(rw,sync,no_root_squash,no_subtree_check)
```

12. Simpan dan restart nfs-kernel-server

    Server

```
kel6@reihan-virtual-machine:~$ sudo nano /etc/exports
kel6@reihan-virtual-machine:~$ sudo exportfs -a
kel6@reihan-virtual-machine:~$ sudo systemctl restart nfs-kernel-server
kel6@reihan-virtual-machine:~$
```

13. Instalasi nfs client

    Client

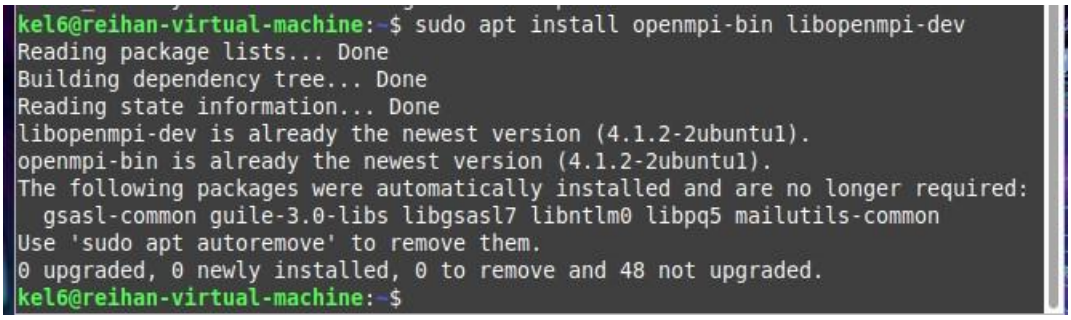
```
kel6@reihanserver:~$ sudo apt install nfs-common
[sudo] password for kel6:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libevent-pthreads-2.1-7 libmecab2 libprotobuf-lite23 mecab-ipadic
  mecab-ipadic-utf8 mecab-utils
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  keyutils libnfsidmap1 rpcbind
Suggested packages:
  watchdog
The following NEW packages will be installed:
  keyutils libnfsidmap1 nfs-common rpcbind
0 upgraded, 4 newly installed, 0 to remove and 5 not upgraded.
```

14. Mounting sharing file pada Client

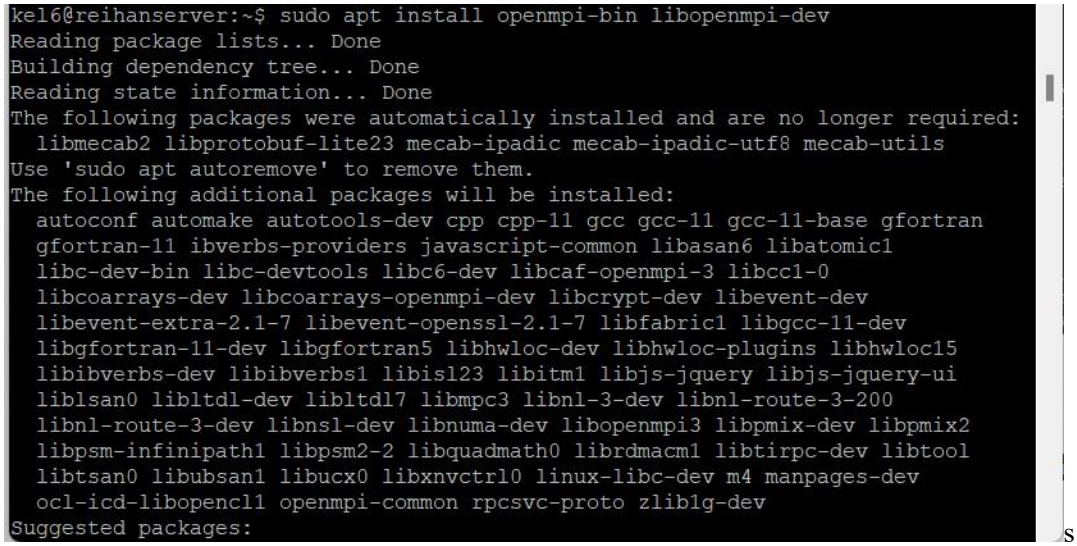
Client

15. Instalasi python3 dan mpi

Server



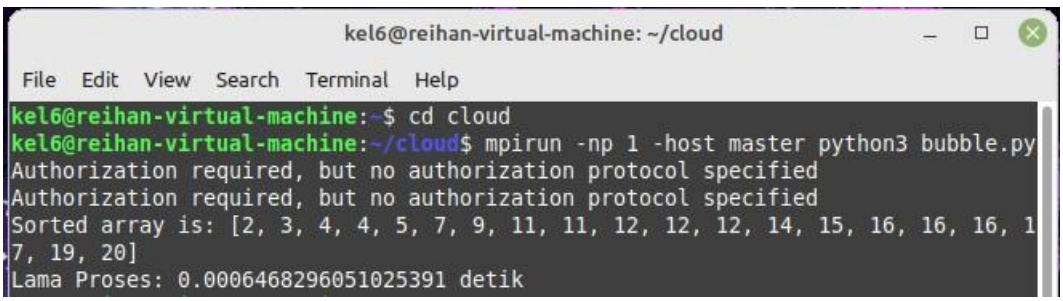
Client



s

16. Menjalankan program bubble sort pada file bubble.py secara multi computing

Server

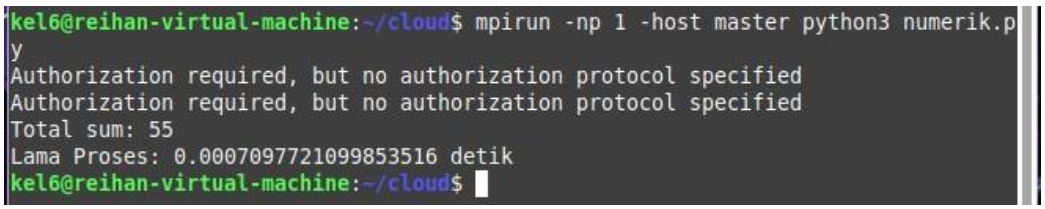


Kodingan :

```
from mpi4py import MPI
import random
import time
```

```
start_time = time.time() def
bubbleSort(arr):     n = len(arr)
swapped = False     for i in range(n - 1):
for j in range(0, n - i - 1):            if arr[j]
> arr[j + 1]:                swapped = True
arr[j], arr[j + 1] = arr[j + 1], arr[j]

       if not swapped:
return

if __name__ == '__main__':
comm = MPI.COMM_WORLD
   size = comm.Get_size()
rank = comm.Get_rank()

   n = 20  # Jumlah elemen dalam array
   max_number = 20  # Rentang angka acak
local_data = []

   # Setiap proses mendapatkan data yang berbeda     for i
in range(n):         local_data.append(random.randint(1,
max_number))     local_data = comm.gather(local_data,
root=0)

   if rank == 0:
      sorted_data = [item for sublist in local_data for item in
sublist]        bubbleSort(sorted_data)        print("Sorted array is:",
sorted_data)

# Waktu selesai end_time
= time.time()

# Hitung lama eksekusi execution_time =
end_time - start_time print(f"Lama Proses:
{execution_time} detik")
```

17. Menjalankan program numeric pada file numerik.py secara multi computing Server



Kodingan :

```
from mpi4py import
MPI import numpy as np
import time start_time =
time.time() def
parallel_sum(data):
comm =
MPI.COMM_WORLD
rank = comm.Get_rank()
size = comm.Get_size()
```

```
    # Bagi data di antara proses      local_data =
np.array_split(data, size)[rank]

    # Hitung jumlah lokal
local_sum = np.sum(local_data)

    # Gather hasil dari setiap proses      total_sum =
comm.reduce(local_sum, op=MPI.SUM, root=0)

    if rank == 0:         print("Total
sum:", total_sum)

if __name__ == '__main__':
    # Data numerik (gunakan data sesuai kebutuhan Anda)
data = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])

    # Panggil fungsi untuk menjalankan program secara paralel
parallel_sum(data)

# Waktu selesai end_time
= time.time()

# Hitung lama eksekusi execution_time =

end_time - start_time print(f"Lama Proses:

{execution_time} detik")
```