

TUGAS KELOMPOK
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 Praktikum Eksekusi Program Bubble Sort Python Menggunakan MPI Secara Paralel

Hal-hal yang diperlukan sebelum menjalankan programnya :

- Pastikan master dan worker/slave sudah dalam satu jaringan atau wifi yang sama.
- Tentukan komputer mana yang akan dijadikan master dan worker/slave, untuk master hanya 1 komputer, untuk worker/slave bisa beberapa komputer atau lebih dari 1.
- Upgrade OS terlebih dahulu dengan menggunakan perintah `sudo apt update && sudo apt upgrade`
- Install beberapa hal berikut; net-tools untuk ngecek IP, vim untuk teks editor. Untuk menginstallnya dapat dilakukan dengan perintah `sudo apt install net-tools vim`.

Pengecekan IP menggunakan `ifconfig`.

Master :

```
rafiriz@rafiriz-virtual-machine:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.0.105  netmask 255.255.255.0  broadcast 192.16
8.0.255
    inet6 fe80::883:2b24:8c:a0d1  prefixlen 64  scopeid 0x20<li
nk>
    ether 00:0c:29:c3:34:92  txqueuelen 1000  (Ethernet)
    RX packets 280384  bytes 250309233 (250.3 MB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 39067  bytes 2374486 (2.3 MB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
    loop txqueuelen 1000  (Local Loopback)
    RX packets 185  bytes 18011 (18.0 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 185  bytes 18011 (18.0 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

Worker 1:

```
rafiriz1@rafiriz1-virtual-machine:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.0.103  netmask 255.255.255.0  broadcast 192.16
8.0.255
    inet6 fe80::e538:a637:138a:8cdc  prefixlen 64  scopeid 0x20
<link>
    ether 00:0c:29:a6:64:e0  txqueuelen 1000  (Ethernet)
    RX packets 7841  bytes 10160414 (10.1 MB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 1972  bytes 131153 (131.1 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

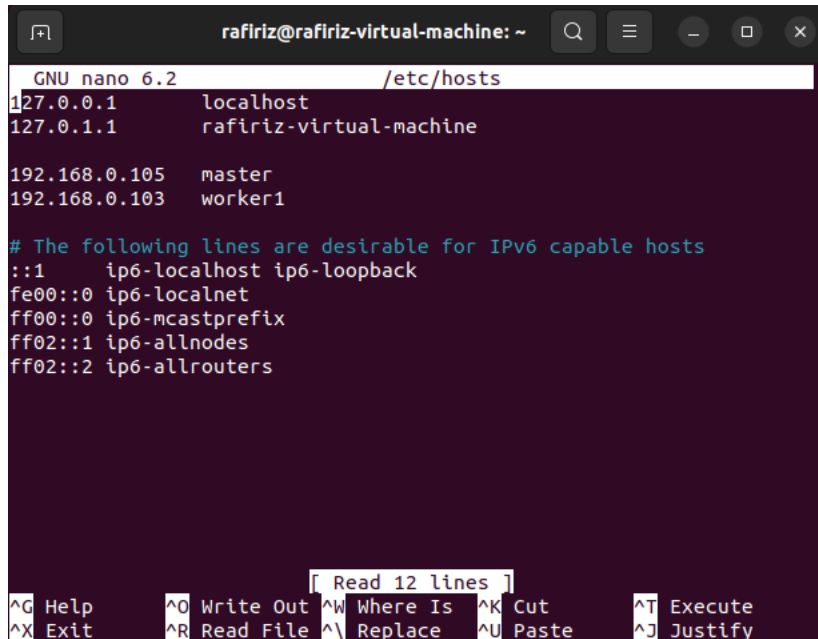
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
    loop txqueuelen 1000  (Local Loopback)
    RX packets 165  bytes 16268 (16.2 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 165  bytes 16268 (16.2 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

rafiriz1@rafiriz1-virtual-machine:~$
```

Proses konfigurasi

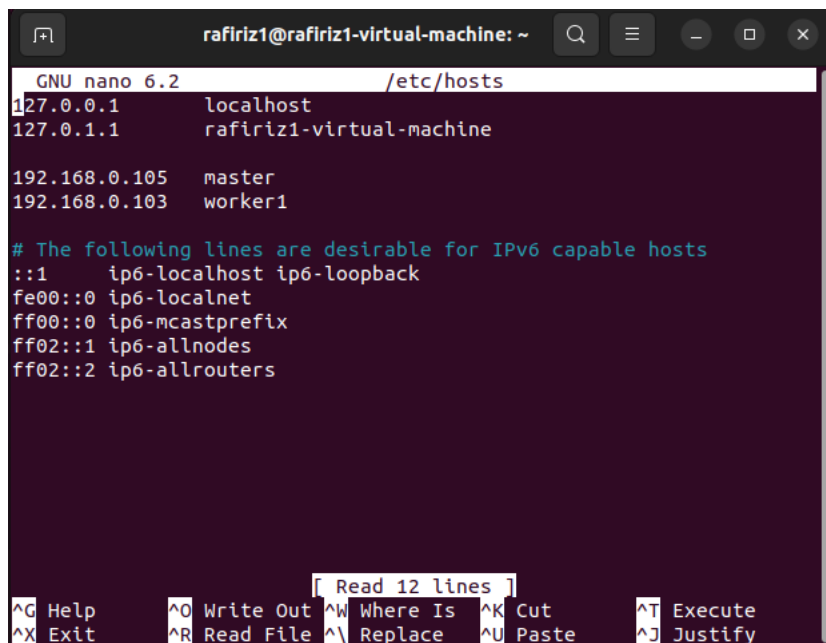
Buka file `/etc/hosts` menggunakan `'sudo nano /etc/hosts'` atau bisa juga menggunakan `'vim /etc/hosts'`, kemudian tambahkan isinya menggunakan ip dari master dan worker 1 diatas tadi.

Master



```
rafiriz@rafiriz-virtual-machine: ~  
GNU nano 6.2 /etc/hosts  
127.0.0.1    localhost  
127.0.1.1    rafiriz-virtual-machine  
  
192.168.0.105 master  
192.168.0.103 worker1  
  
# The following lines are desirable for IPv6 capable hosts  
::1        ip6-localhost ip6-loopback  
fe00::0    ip6-localnet  
ff00::0    ip6-mcastprefix  
ff02::1    ip6-allnodes  
ff02::2    ip6-allrouters  
  
[ Read 12 lines ]  
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  
^X Exit      ^R Read File ^_ Replace   ^U Paste     ^J Justify
```

Worker 1



```
rafiriz1@rafiriz1-virtual-machine: ~  
GNU nano 6.2 /etc/hosts  
127.0.0.1    localhost  
127.0.1.1    rafiriz1-virtual-machine  
  
192.168.0.105 master  
192.168.0.103 worker1  
  
# The following lines are desirable for IPv6 capable hosts  
::1        ip6-localhost ip6-loopback  
fe00::0    ip6-localnet  
ff00::0    ip6-mcastprefix  
ff02::1    ip6-allnodes  
ff02::2    ip6-allrouters  
  
[ Read 12 lines ]  
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  
^X Exit      ^R Read File ^_ Replace   ^U Paste     ^J Justify
```

Membuat User Baru

Membuat user baru untuk master dan worker 1 menggunakan perintah 'sudo adduser <nama user>' dan di sini saya menggunakan nama uaskelpp.

```
rafiriz@rafiriz-virtual-machine:~$ sudo adduser uaskelpp
Adding user 'uaskelpp' ...
Adding new group 'uaskelpp' (1002) ...
Adding new user 'uaskelpp' (1002) with group 'uaskelpp' ...
Creating home directory '/home/uaskelpp' ...
Copying files from '/etc/skel' ...
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
Changing the user information for uaskelpp
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
```

Kemudian modifikasi user dengan menggunakan perintah pertama dibawah, dan pindah ke pengguna yang telah dibuat tadi menggunakan perintah 'su -'.

```
rafiriz@rafiriz-virtual-machine:~$ sudo usermod -aG sudo uaskelpp
rafiriz@rafiriz-virtual-machine:~$ su - uaskelpp
Password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

uaskelpp@rafiriz-virtual-machine:~$
```

Konfigurasi SSH

Jika anda belum menginstall ssh gunakan perintah 'sudo apt install openssh-server'. Setelah terinstall, lakukan pengecekan ssh dengan menggunakan perintah 'ssh <namauser>@<host>'. Host disesuaikan dengan master apabila komputer master dan dan worker1 apabila worker 1.

Master

```
uaskelpp@rafiriz-virtual-machine:~$ ssh uaskelpp@master
The authenticity of host 'master (192.168.0.105)' can't be established.
ED25519 key fingerprint is SHA256:uywP/W+H+T04hov7PB8foj+1TtEEiZBTzYFxuCOZsM8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'master' (ED25519) to the list of known hosts.
uaskelpp@master's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-35-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Expanded Security Maintenance for Applications is not enabled.
```

Worker 1

```
uaskelpp@rafiriz1-virtual-machine:~$ ssh uaskelpp@worker1
The authenticity of host 'worker1 (192.168.0.103)' can't be established.
ED25519 key fingerprint is SHA256:UVC6PgFhHofQ+ORRQImuE/pj6/NQRAQzg
mPA1LFXUW0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])
? yes
Warning: Permanently added 'worker1' (ED25519) to the list of known
hosts.
uaskelpp@worker1's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-36-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Expanded Security Maintenance for Applications is not enabled.
```

Kemudian jalankan perintah ‘ssh-keygen -t rsa’ di master untuk menghasilkan pasangan kunci SSH dengan algoritma enkripsi RSA.

```
uaskelpp@rafiriz-virtual-machine:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/uaskelpp/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/uaskelpp/.ssh/id_rsa
Your public key has been saved in /home/uaskelpp/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:PAXImI7U+F6FR7l33N6XXLNCUbKqD01ayeJ9Mmjfn/k uaskelpp@rafiriz
-virtual-machine
The key's randomart image is:
+---[RSA 3072]---+
|  o + +o.      + |
|  o + + +.    o  |
| . +   o.o. .  o |
| . o ...o.o.o .oo|
| . .   S.=...+.* |
|   .   o B ..=o  |
|   .   * . .    |
|   = +.. o      |
|   . ..+...+E   |
+-----[SHA256]-----+
```

Kemudian copas isi dari file ‘id_rsa.pub’ ke ‘authorized_keys’ dengan menggunakan perintah seperti dibawah ini.

```
uaskelpp@rafiriz-virtual-machine:~$ cd .ssh
uaskelpp@rafiriz-virtual-machine:~/.ssh$ cat id_rsa.pub | ssh uaske
lpp@master "mkdir .ssh; cat >> .ssh/authorized_keys"
uaskelpp@master's password:
mkdir: cannot create directory '.ssh': File exists
uaskelpp@rafiriz-virtual-machine:~/.ssh$ cat id_rsa.pub | ssh uaske
lpp@worker1 "mkdir .ssh; cat >> .ssh/authorized_keys"
The authenticity of host 'worker1 (192.168.0.103)' can't be established.
ED25519 key fingerprint is SHA256:UVC6PgFhHofQ+ORRQImuE/pj6/NQRAQzg
mPA1LFXUW0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])
? yes
Warning: Permanently added 'worker1' (ED25519) to the list of known
hosts.
uaskelpp@worker1's password:
mkdir: cannot create directory '.ssh': File exists
uaskelpp@rafiriz-virtual-machine:~/.ssh$ ls
authorized keys  id_rsa  id_rsa.pub  known hosts  known hosts.old
```

Konfigurasi NFS

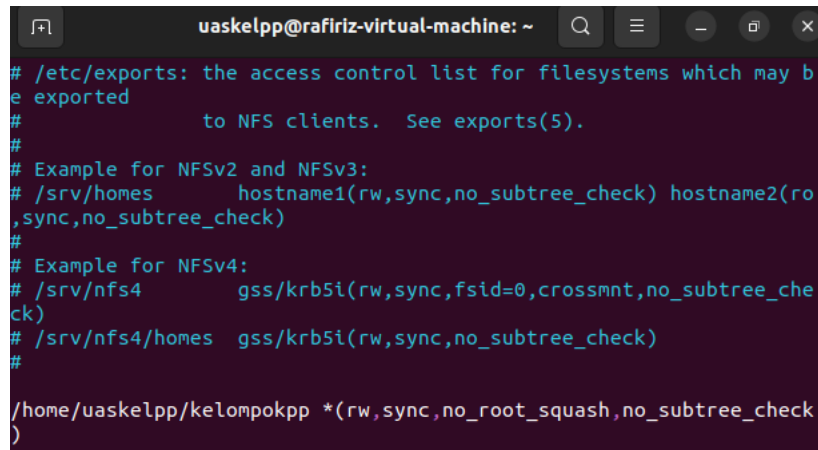
Disini buat pada master dan worker 1 folder baru bernama 'kelompokpp'.

```
uaskelpp@rafiriz-virtual-machine:~$ mkdir kelompokpp
```

Kemudian install nfs kernel server pada master saja.

```
uaskelpp@rafiriz-virtual-machine:~$ sudo apt install nfs-kernel-server
```

Setelah itu buka file menggunakan perintah 'vim /etc/exports' dan ubah isinya seperti dibawah ini.



```
uaskelpp@rafiriz-virtual-machine: ~  
# /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_subtree_check)  
#  
# 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/uaskelpp/kelompokpp *(rw,sync,no_root_squash,no_subtree_check)
```

Kemudian sebelum menjalankan perintah dibawah, install terlebih dahulu nfs common. Jika sudah maka lakukan perintah.

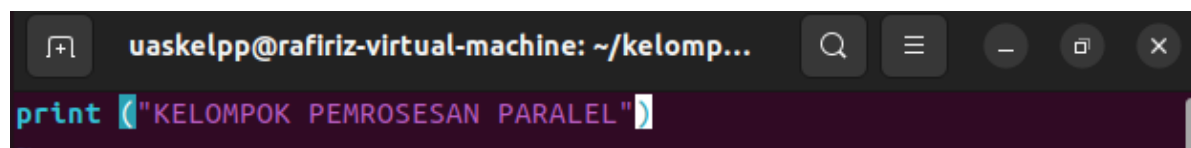
```
uaskelpp@rafiriz-virtual-machine:~$ sudo exportfs -a  
uaskelpp@rafiriz-virtual-machine:~$ sudo systemctl restart nfs-kernel-server
```

Selanjutnya lakukan mounting atau mengaitkan sistem file jarak jauh yang dibagikan dari master ke lokasi tertentu pada komputer worker 1.

```
uaskelpp@rafiriz1-virtual-machine:~$ sudo mount master:/home/uaskelpp/kelompokpp /home/uaskelpp/kelompokpp
```

Proses Testing MPI

Install MPI terlebih dahulu menggunakan perintah 'sudo apt install openmpi-bin libopenmpi-dev'. Gunakan perintah touch untuk membuat file python tersebut, disini buat file python ngetest.py dan membuka isinya dengan menggunakan perintah vim, kemudian mengisinya seperti dibawah ini.



```
uaskelpp@rafiriz-virtual-machine: ~/kelomp...  
print ("KELOMPOK PEMROSESAN PARALEL")
```

Kemudian run file python tersebut dengan menggunakan perintah dibawah ini.

```
maskelpp@rafiriz-virtual-machine:~/kelompokpp$ mpirun -np 2 -host m
aster,worker1 python3 ngetest.py
Authorization required, but no authorization protocol specified
Authorization required, but no authorization protocol specified
Authorization required, but no authorization protocol specified
Authorization required, but no authorization protocol specified
Authorization required, but no authorization protocol specified
Authorization required, but no authorization protocol specified
Authorization required, but no authorization protocol specified
Authorization required, but no authorization protocol specified
KELOMPOK PEMROSESAN PARALEL
KELOMPOK PEMROSESAN PARALEL
```

Eksekusi Program Bubble Sort Python Menggunakan MPI Secara Paralel

Install terlebih dahulu python dan mpi4py, kemudian masukkan kode di python tersebut seperti dibawah ini.

```
from mpi4py import MPI

def parallel_bubble_sort(data, comm):
    rank = comm.Get_rank()
    size = comm.Get_size()
    local_data = data[rank::size] # Pisahkan data menjadi bagian yang sama untuk setiap prosesor

    for _ in range(size):
        local_data = local_bubble_sort(local_data) # Urutkan data lokal dengan Bubble Sort

        if rank % 2 == 0:
            if rank < size - 1:
                comm.send(local_data, dest=rank + 1)
                received_data = comm.recv(source=rank + 1)
                local_data = merge(local_data, received_data)
            else:
                if rank > 0:
                    received_data = comm.recv(source=rank - 1)
                    comm.send(local_data, dest=rank - 1)
                    local_data = merge(local_data, received_data)

    return local_data

def local_bubble_sort(data):
    n = len(data)
    for i in range(n):
        for j in range(0, n-i-1):
            if data[j] > data[j+1]:
                data[j], data[j+1] = data[j+1], data[j]
    return data

def merge(left, right):
    result = []
    i = j = 0
    while i < len(left) and j < len(right):
        if left[i] < right[j]:
            result.append(left[i])
            i += 1
        else:
            result.append(right[j])
            j += 1
```

```
    result += left[i:]
    result += right[j:]
    return result

if __name__ == "__main__":
    comm = MPI.COMM_WORLD
    rank = comm.Get_rank()

    if rank == 0:
        data = [5, 2, 9, 3, 1, 8, 6, 4, 7]
        sorted_data = parallel_bubble_sort(data, comm)
        print("Sorted Data:", sorted_data)
    else:
        parallel_bubble_sort([], comm)
```

Dari kode diatas, perintah-perintah tersebut berfungsi seperti penjelasan dibawah ini :

1. 'from mpi4py import MPI' : Mengimpor pustaka MPI (mpi4py) untuk digunakan dalam program MPI.
2. 'def parallel_bubble_sort(data, comm)' : Fungsi utama untuk mengurutkan data secara paralel. Itu menerima dua parameter: data adalah daftar yang akan diurutkan, dan comm adalah objek komunikator MPI.
3. 'rank = comm.Get_rank()' : Mengambil nomor identitas dari prosesor saat ini dalam komunikator MPI.
4. 'size = comm.Get_size()' : Mengambil jumlah total prosesor dalam komunikator MPI.
5. 'local_data = data[rank::size]' : Membagi data ke dalam bagian yang sama untuk setiap prosesor. Setiap prosesor hanya bekerja dengan bagian yang sesuai dari data.
6. 'for _ in range(size)' : Loop utama yang akan diulang sebanyak prosesor yang terlibat. Ini diperlukan agar setiap prosesor dapat melakukan pengurutan dan pertukaran data secara bersamaan.
7. 'local_data = local_bubble_sort(local_data)' : Memanggil fungsi local_bubble_sort untuk mengurutkan data lokal pada setiap prosesor menggunakan algoritma Bubble Sort.
8. Blok berikutnya menangani pertukaran data antar prosesor (merger). Pertukaran data terjadi di antara prosesor dengan nomor genap (0, 2, 4, dst.) dan genap (1, 3, 5, dst.).
 - Jika nomor prosesor adalah genap ($\text{rank} \% 2 == 0$), maka data di-send ke prosesor berikutnya ($\text{dest}=\text{rank}+1$) dan menerima data dari prosesor tersebut.
 - Jika nomor prosesor adalah ganjil, maka menerima data dari prosesor sebelumnya ($\text{source}=\text{rank}-1$) dan mengirimkan data ke prosesor tersebut.
 - Setelah pertukaran, data di-merge menggunakan fungsi merge().
9. 'return local_data' : Fungsi ini mengembalikan data lokal yang sudah diurutkan oleh masing-masing prosesor.
10. 'def local_bubble_sort(data)' : Fungsi yang mengurutkan data lokal menggunakan algoritma Bubble Sort.
11. 'for i in range(n)', 'for j in range(0, n-i-1)' : Loop bersarang yang digunakan untuk menjalankan Bubble Sort pada data lokal. Bubble Sort membandingkan elemen-elemen berdampingan dan menukar jika diperlukan untuk mengurutkannya.
12. 'merge(left, right)' : Fungsi untuk menggabungkan dua daftar terurut (dalam kasus ini, hasil pengurutan lokal dari dua prosesor yang berdekatan). Ini menggunakan teknik "merge" standar dalam penggabungan data terurut.
13. 'if __name__ == "__main__"' : Ini memulai eksekusi program jika program dieksekusi sebagai skrip utama (bukan sebagai modul yang diimpor).
14. `comm = MPI.COMM_WORLD, rank = comm.Get_rank()`: Ini menginisialisasi komunikator MPI (comm) dan mendapatkan nomor identitas prosesor saat ini (rank) dalam komunikator tersebut.
15. Blok 'if rank == 0:' : Di sini, prosesor dengan nomor 0 adalah prosesor utama yang menginisialisasi data dan menampilkan hasil pengurutan.
16. Blok 'else' : Prosesor lain akan menjalankan fungsi parallel_bubble_sort dan melakukan pekerjaan pengurutan, tetapi mereka tidak menampilkan hasil pengurutan.

Yang terakhir output yang didapat dari kode python tersebut adalah seperti yang tertera dibawah ini.

```
uaskelpp@rafiriz-virtual-machine:~/kelompokpp$ mpirun -np 2 -host master,worker1 python3 ngetest.py
Authorization required, but no authorization protocol specified
Authorization required, but no authorization protocol specified
Authorization required, but no authorization protocol specified
Authorization required, but no authorization protocol specified
Authorization required, but no authorization protocol specified
Authorization required, but no authorization protocol specified
Authorization required, but no authorization protocol specified
Sorted Data: [1, 1, 5, 5, 6, 6, 7, 7, 9, 9]
```