LAPORAN PRAKTIKUM ARSIKOM

(Arsitektur dan Organisasi Komputer)

Disusun Oleh :

1. **Zinniarethie Andari Kostiene (09030582226007)**
2. **Sachio Aji (09030582226043)**
3. **Pio Genty Zahir (09030582226045)**
4. **Wahyu Hidayat (09030582226000)**

Prodi : Teknik Komputer

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

LABORATORIUM PERANGKAT KERAS

FAKULTAS ILMU KOMPUTER

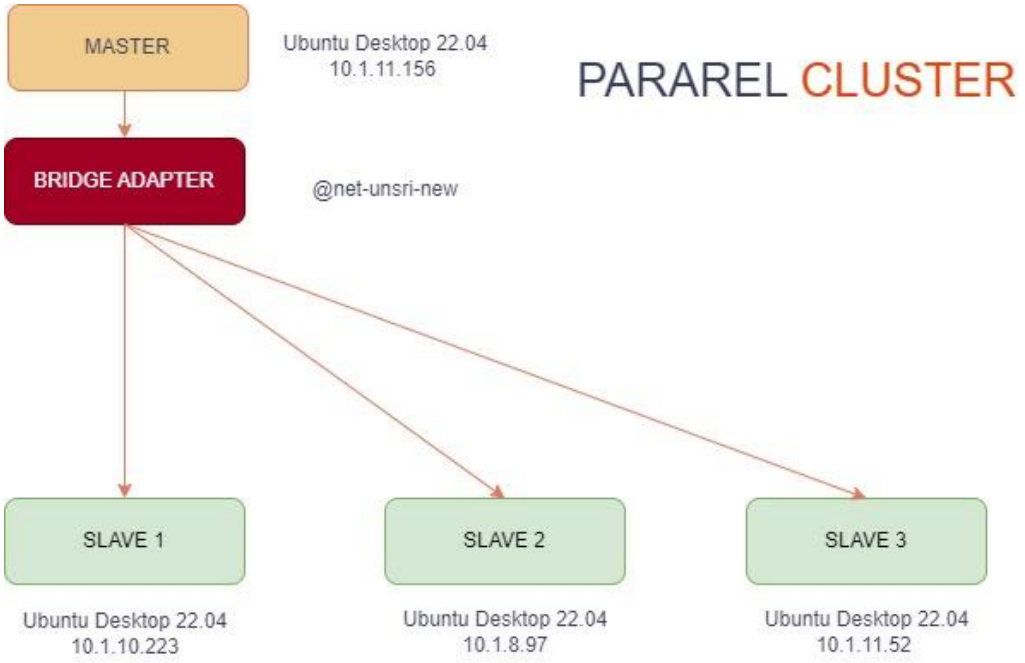
UNIVERSITAS SRIWIJAYA

PALEMBANG 2023

Device Dan Tools Yang di gunakan:

1. Ubuntu Desktop
   - Ubuntu Desktop Master
   - Ubuntu Desktop Slave 1
   - Ubuntu Desktop Slave 2
   - Ubuntu Desktop Slave 3
2. MPI (Master dan Slave)
3. SSH (Master dan Slave)
4. Codingan numerik python

Topology :



Pada percobaan kali ini digunakan empat komputer, dimana satu komputer sebagai master, dan tiga komputer lainnya sebagai slave. Pastikan sudah berada dalam satu jaringan yang sama.

Kemudian pada pengeksekusian ini terdapat komputer yaitu master (pipop-VirtualBox), dan 3 komputer lainnya yaitu (dayat-VirtualBox) sebagai slave1, (pio-VirtualBox) sebagai slave2, (zizi) sebagai slave3. Dan pengeksekusian ini bertujuan untuk computing.

- **Konfigurasi file /etc/hosts**

Lakukan pada master dan slave:

Edit file `/etc/hosts` melalui nano. Tambahkan isinya dengan beberapa IP dan aliasny. Di bawahini sebagai contoh. sesuaikan IP nya dengankomputer masing-masing.Untuk mengecek IP gunakan perintah `ifconfig`.

Tambahkan baris berikut dengan format

[10.1.11.156] [pipop-VirtualBox]
[10.1.10.223] [dayat-VirtualBox]
[10.1.8.97] [pio-VirtualBox]
[10.1.11.52] [zizi]

Lakukan disetiap komputer.

Master:

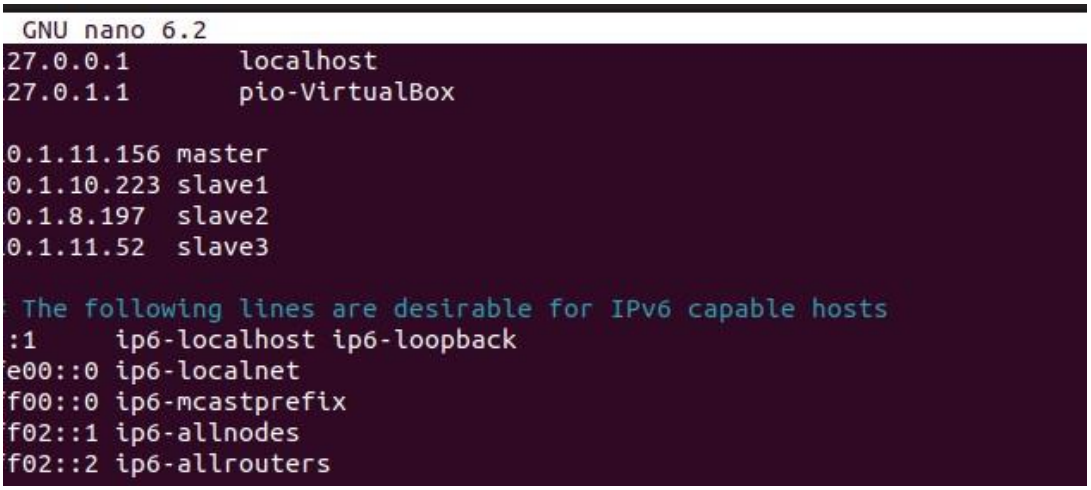
```
127.0.0.1       localhost
127.0.1.1       pipop-VirtualBox


10.1.11.156 master
10.1.10.223 slave1
10.1.8.197  slave2
10.1.11.52  slave3
# 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
```

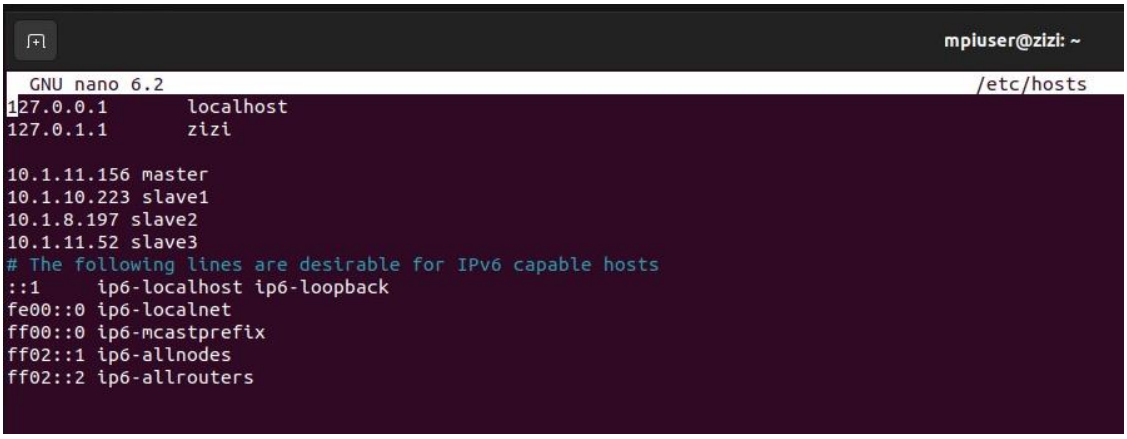
Slave1:

```
  GNU nano 6.2                          /etc/hosts
127.0.0.1       localhost
127.0.1.1       dayat-VirtualBox
10.1.11.156 master
10.1.10.233 slave1
10.1.8.197 slave2
10.1.11.52 slave3
# 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
```

Slave2:


GNU nano 6.2
27.0.0.1 localhost
27.0.1.1 pio-VirtualBox
0.1.11.156 master
0.1.10.223 slave1
0.1.8.197 slave2
0.1.11.52 slave3
The following lines are desirable for IPv6 capable hosts
:1 ip6-localhost ip6-loopback
e00::0 ip6-localnet
f00::0 ip6-mcastprefix
f02::1 ip6-allnodes
f02::2 ip6-allrouters

Slave3:



- **Buat User Baru**

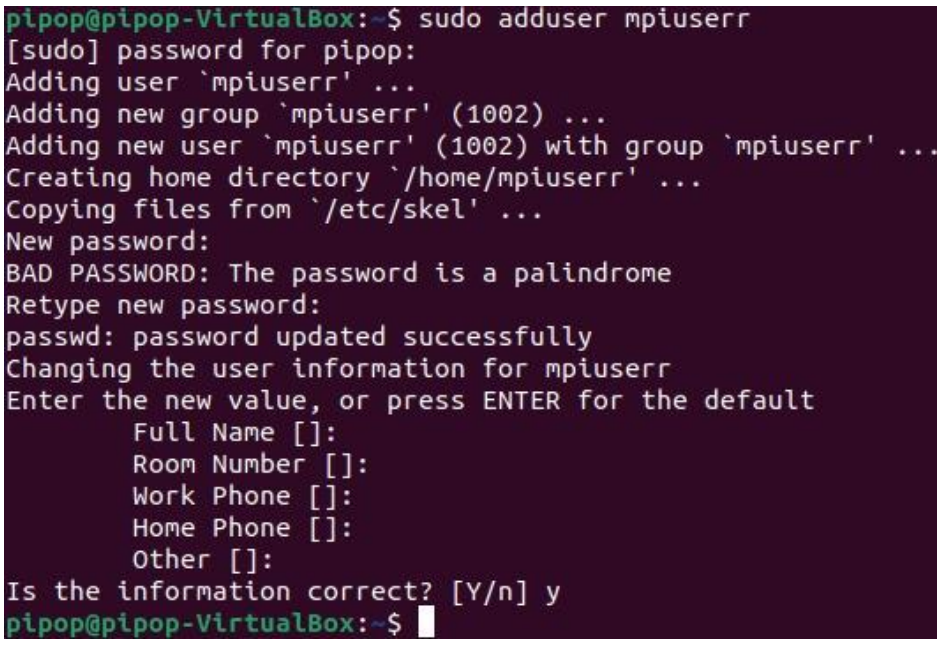
1. **Buat user**

Buat user baru di master dan slave dengan perintah berikut:

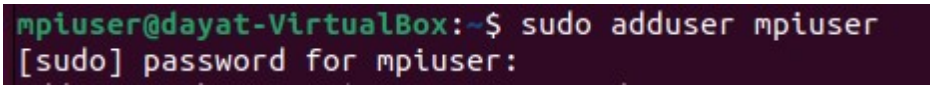
```
Sudo adduser <nama user>
```

Pada percobaan ini digunakan nama user mpiuse r. Nama user harus sama pada kompuer master dan slave.

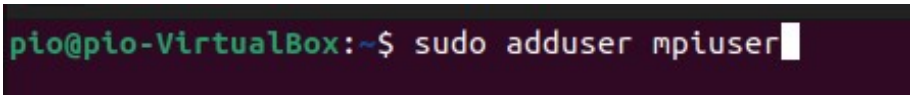
Master:



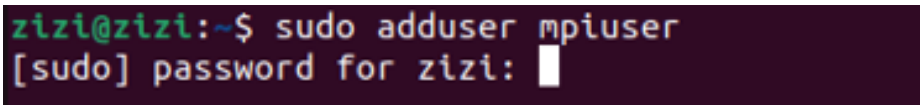
Slave1:



Slave2:



Slave3:



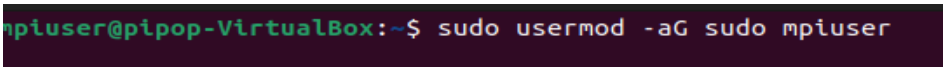
2. Memberiakses root ke user

Lakukan di master dan slave dengan perintah berikut:

```
Sudo usermod –aG sudo mpiuser
```

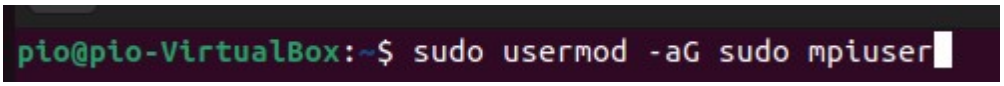
Perintah ini akan menjadikan user `mpiuser` akan memiliki hak akses superuser.

Master:

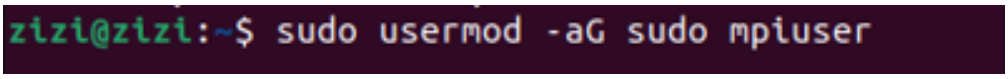


Slave1:

Slave2:



Slave3:

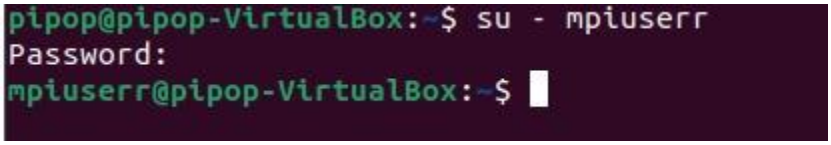


3. **Masuk ke user**

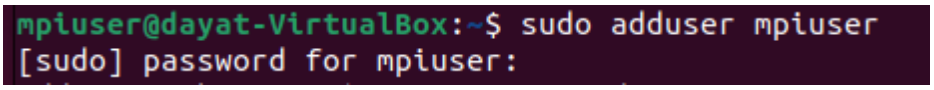
Lakukan di master dan slave dengan perintah berikut:

```
su – mpiuser
```

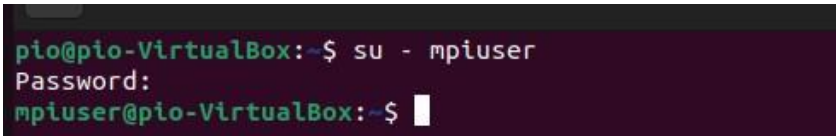
Master:



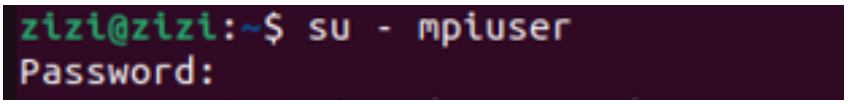
Slave1:



Slave2:



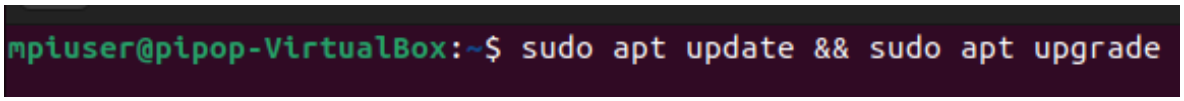
Slave3:



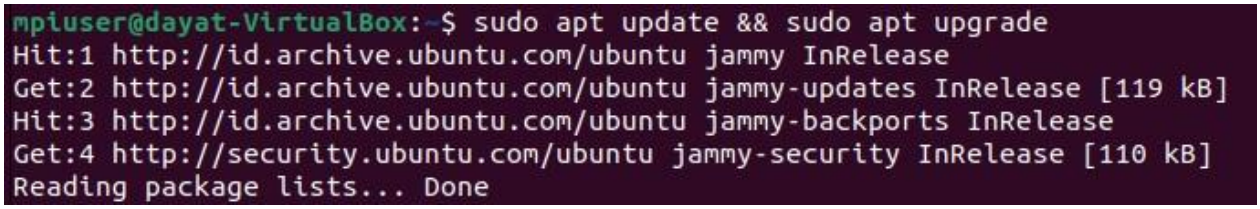
- **Meng update Ubuntu Dekstop dan Menginstall tools**

1. **Update dan Upgrade**

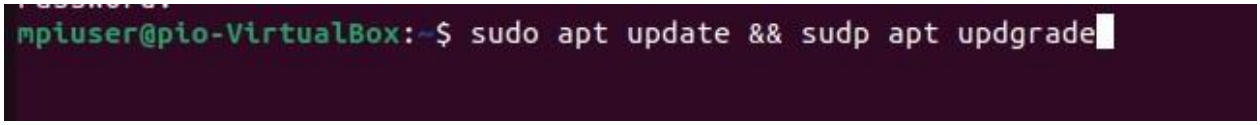
Master:



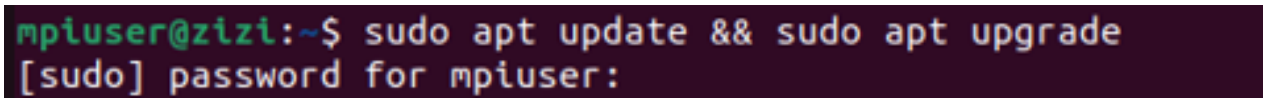
Slave1:



Slave2:

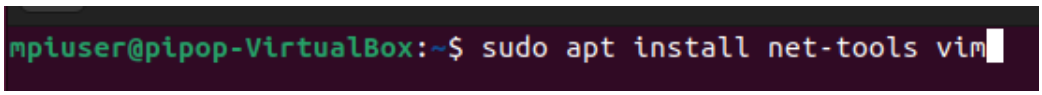


Slave3:

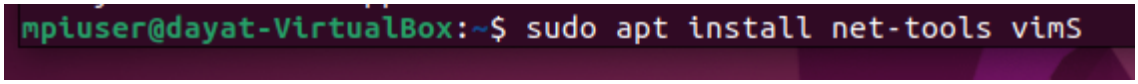


2. **Install net-tools**

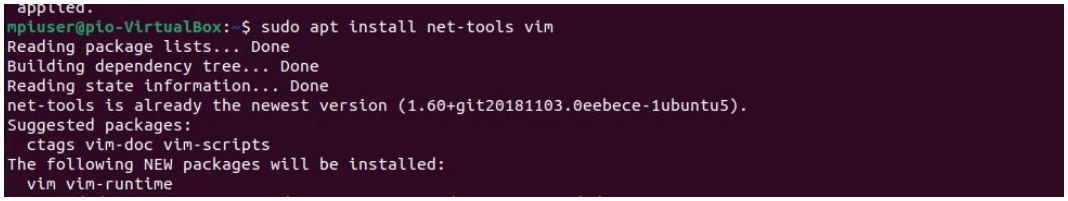
Master:



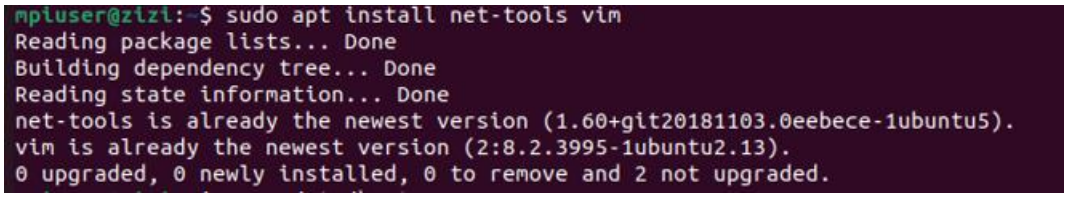
Slave1:



Slave2:

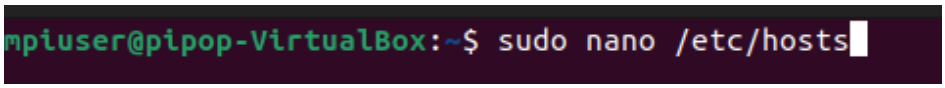


Slave3:

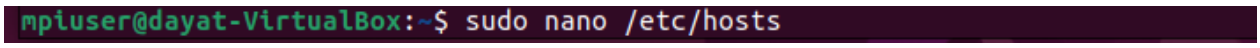


- **Konfigurasikan file pada master, slave1, slave2, dan slave3. Daftarkan IP dan nama host setiap komputer**

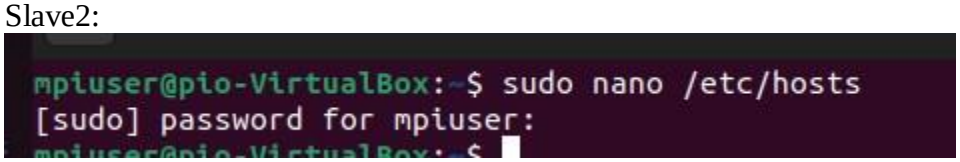
Master:



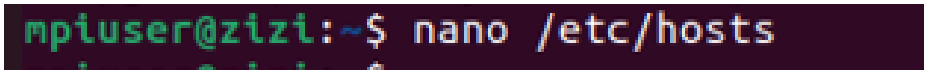
Slave1:



Slave2:



Slave3:



- **Konfigurasi SSH**

Setelah masuk ke user, lakukan konfigurasi SSH.

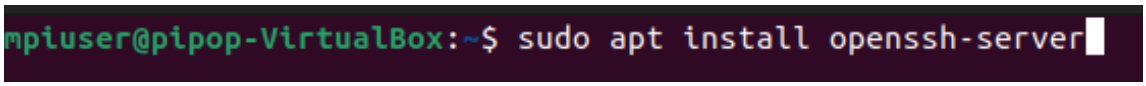
1. **Install SSH**

Lakukan di master dan slave:

```
sudo apt install openssh-server
```

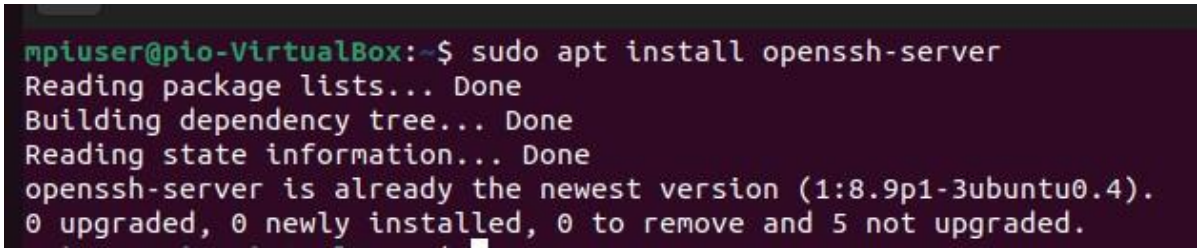
Perintah tersebut akan menginstal perangkat lunak OpenSSH Server pada sistem jika belum terinstal, sehingga Anda dapat menggunakan layanan SSH untuk mengakses dan mengelola sistem secara remote dengan aman.

Master:



Slave1:

Slave2:



Slave3:

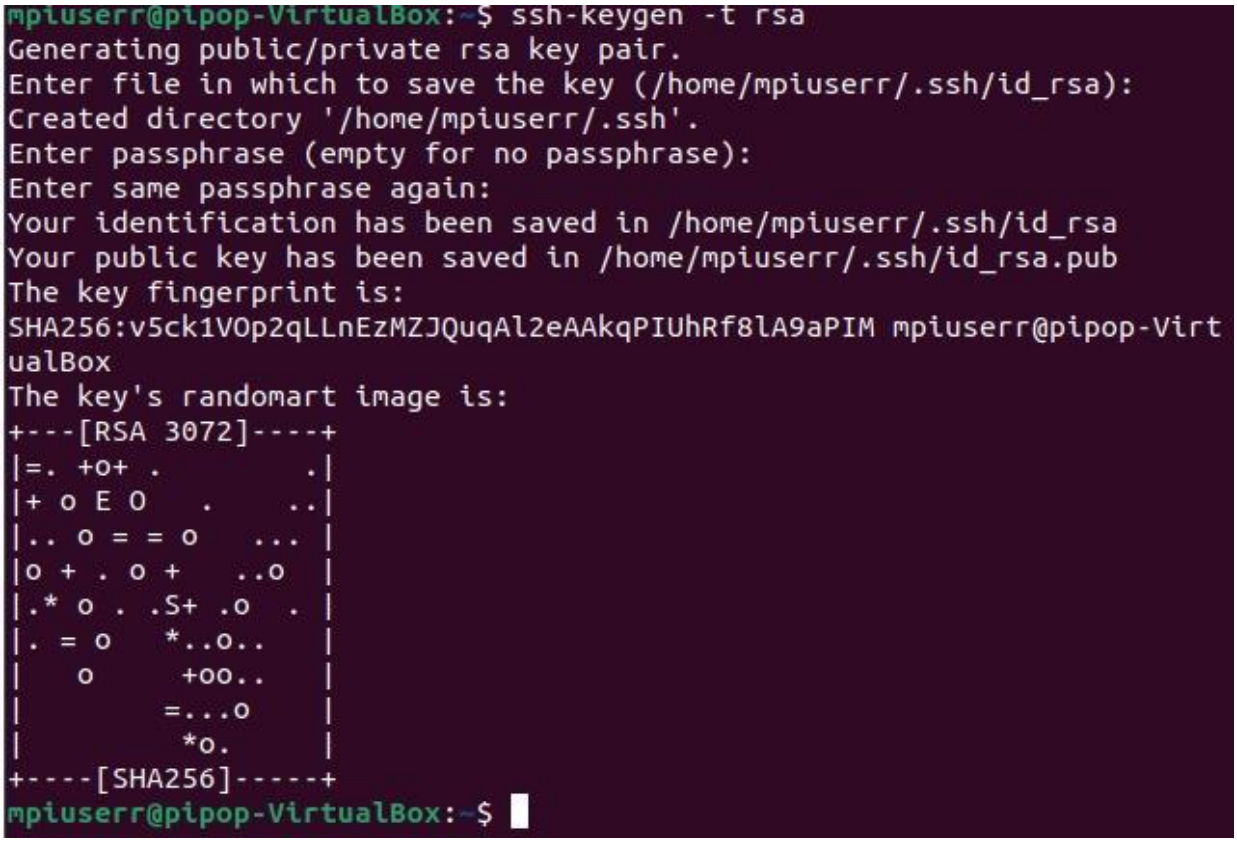
2. Generate Keygen

Lakukan di master:

```
Ssh-keygen -t rsa
```

Perintah ini akan membuat kunci SSH baru.

Master:


mpiuserr@pipop-VirtualBox:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/mpiuserr/.ssh/id_rsa):
Created directory '/home/mpiuserr/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/mpiuserr/.ssh/id_rsa
Your public key has been saved in /home/mpiuserr/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:v5ck1VOp2qLLnEzMZJQuqAl2eAAkqPIUhRf8lA9aPIM mpiuserr@pipop-Virt
ualBox
The key's randomart image is:
+---[RSA 3072]----+
|=. +o+ .   .|
|+ o E O   .   ..|
|.. o = = o   ... |
|o + . o +   ..o |
|.* o . .S+ .o . |
|. = o   *..o.. |
| o   +oo.. |
| =...o |
| *o. |
+----[SHA256]-----+
mpiuserr@pipop-VirtualBox:~$

3. Copy key publikkeslave

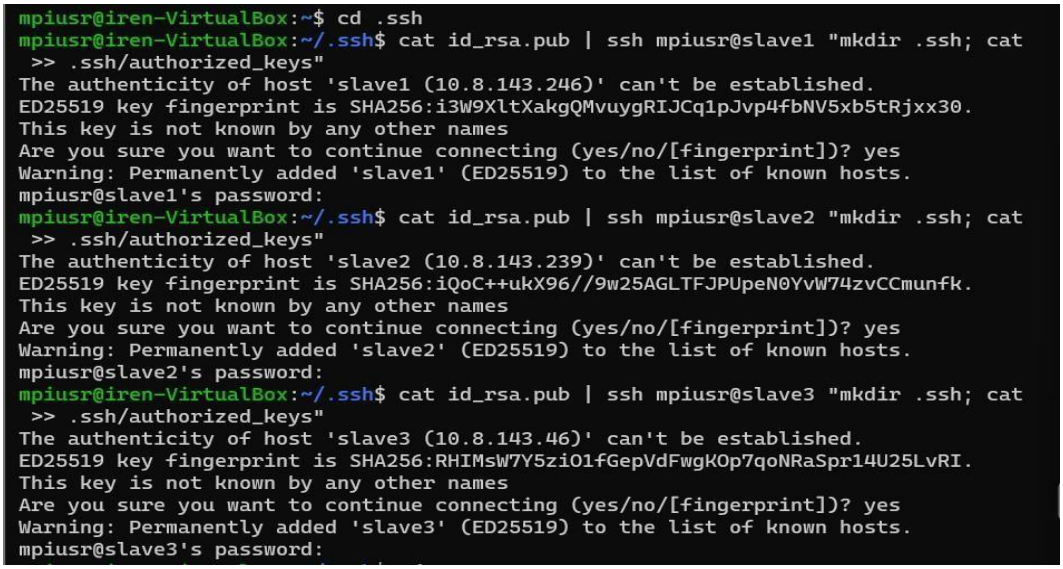
Lakukan di master:

Salin isidari file id_rsa.pubke file authorized_keyske semua slave menggunakan perintah berikut:

```
cd .ssh
cat id_rsa.pub | ssh <nama user>@<host>"mkdir .ssh; cat >>
.ssh/authorized_keys"
```

Lakukan penyalinan perintah berulang-ulang dari master ke slave dengan mengubah <host> menjadi nama host masing-masing slave.

master:



Dengan membagikan kunci SSH, master akan dapat mengakses server slave jarak jauh dengan aman tanpa perlu memasukkan kata sandi setiap kali.

- **Konfigurasi NFS**

1. **Buat shared folder**

   Lakukan di master dan slave dengan perintah mkdir :

   ```
   mkdir /home/mpiuser/bubble
   ```

   master:

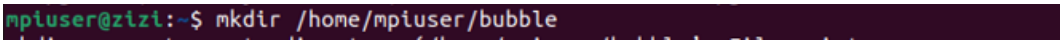
   Slave1:

   Slave2:

   Slave3:

2. **Install NFS Server**

   Lakukan di master:

   ```
   sudo apt install nfs-kernel-server
   ```

   Perintah ini akan menginstall paket `nfs-kernel-server` pada master agar dapat berbagi direktori atau sistem berkas dengan slave.

3. **Konfigurasi file /etc/exports**

   Lakukan di master:

   Edit file `/etc/exports` dengan editor nano

   ```
   sudonano /etc/exports
   ```

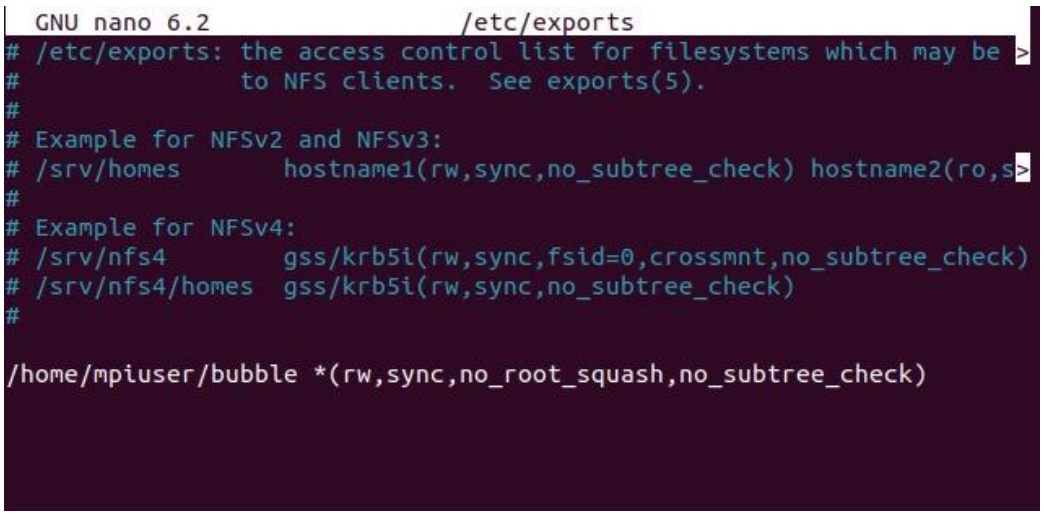
   tambahkan baris berikut.

<lokasi shared folder> *(rw,sync,no_root_squash,no_subtree_check)

Sesuaikan <lokasi shared folder> dengan lokasi folder yang telah dibuat:

```
/home/mpiusr/wa
*(rw,sync,no_root_squash,no_subtree_check)
```

Master:



Lakukan perintah berikut untuk memastikan bahwa perubahan konfigurasi yang dilakukan dalam file `/etc/exports` diterapkan tanpa harus memulai ulang layanan NFS.

```
Sudo exportfs -a
```

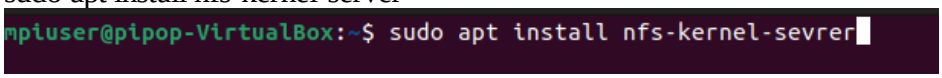
Jalankan perintah ini untuk memuat ulang layanan server NFS dan menerapkan perubahan konfigurasi terbaru dalam file konfigurasi `/etc/exports`.

```
Sudo systemctl restart nfs-kernel-server
```

4. Install NFS

Lakukan di Master dengan perintah berikut:

sudo apt install nfs-kernel-server

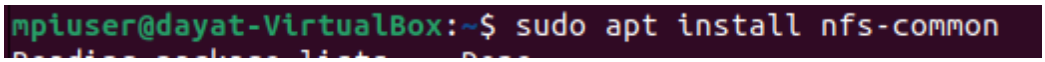


Lakukan di slave dengan perintah berikut:

```
sudo apt install nfs-common
```

Paket nfs-common akan diinstall, memungkinkan untuk mengakses dan menggunakan berkas yang dibagikan oleh master NFS yang telah dikonfigurasi dengan benar.

Slave1:



Slave2:

```
mpiuser@pio-VirtualBox:~$ sudo apt install nfs-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nfs-common is already the newest version (1:2.6.1-1ubuntu1.2).
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
mpiuser@pio-VirtualBox:~$
```

Slave3:

```
mpiuser@zizi:~$ sudo apt install nfs-common
[sudo] password for mpiuser:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nfs-common is already the newest version (1:2.6.1-1ubuntu1.2).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
```

5. **Mounting**

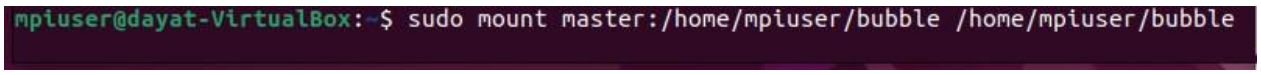
   Lakukan di slave dengan perintah berikut:

```
sudo mount <server host>:<lokasi shared folder di master><lokasi shared folder di slave>
```

sesuaikan <server host>, <lokasi shared folder di master> dan<lokasi shared folder di slave>

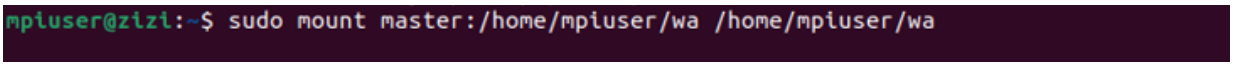
```
sudo mount master:/home/mpiusr/fix /home/mpiusr/fix
```

Slave1:



Slave2:

Slave3:



6. Mpi Installation

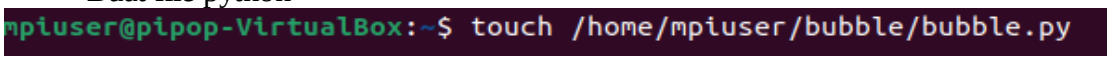
Install open MPI di Master dan Seluruh Slave, dengan command
"Sudo apt install openmpi-bin libopenmpi-dev"

Install the MPI library via pip

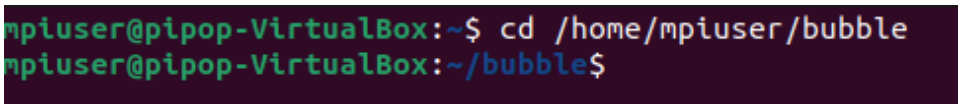
Dengan Command sudo apt install python3-pip
"Pip install mpi4py"

7. Running python Code

Buat file python



Masuk ke directory bubble



Masuk ke file python bubble.py dengan command sudo nano bubble.py
Dan masukkan code pythonnya

```
from mpi4py import MPI

def parallel_bubble_sort(arr):
    comm = MPI.COMM_WORLD
    rank = comm.Get_rank()
    size = comm.Get_size()

    local_arr = arr[rank::size]

    for i in range(len(local_arr)):
        for j in range(0, len(local_arr) - i - 1):
            if local_arr[j] > local_arr[j + 1]:
                local_arr[j], local_arr[j + 1] = local_arr[j + 1], local_arr[j]

    sorted_arr = comm.gather(local_arr, root=0)
```

```
    if rank == 0:
        combined_arr = [item for sublist in sorted_arr for item in sublist]
        combined_arr.sort()
        for i in range(len(arr)):
            arr[i] = combined_arr[i]

if _name_ == '_main_':
    comm = MPI.COMM_WORLD
    rank = comm.Get_rank()

    if rank == 0:
        arr = [5, 3, 4, 1, 2]
    else:
        arr = None

    arr = comm.bcast(arr, root=0)  # Broadcast the arr from rank 0 to all nodes

    comm.barrier()

    parallel_bubble_sort(arr)

    if rank == 0:
            print(f"List sorted with bubble sort in ascending order: {arr}")
```

```
 GNU nano 6.2                                    bubble.py
from mpi4py import MPI

def parallel_bubble_sort(arr):
    comm = MPI.COMM_WORLD
    rank = comm.Get_rank()
    size = comm.Get_size()

    local_arr = arr[rank::size]

    for i in range(len(local_arr)):
        for j in range(0, len(local_arr) - i - 1):
            if local_arr[j] > local_arr[j + 1]:
                local_arr[j], local_arr[j + 1] = local_arr[j + 1], local_arr[j]

    sorted_arr = comm.gather(local_arr, root=0)

    if rank == 0:
        combined_arr = [item for sublist in sorted_arr for item in sublist]
        combined_arr.sort()
        for i in range(len(arr)):
            arr[i] = combined_arr[i]

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

    if rank == 0:
        arr = [5, 3, 4, 1, 2]
    else:
        arr = None

    arr = comm.bcast(arr, root=0)  # Broadcast the arr from rank 0 to all nodes

    comm.barrier()

    parallel_bubble_sort(arr)

    if rank == 0:
        print(f"List sorted with bubble sort in ascending order: {arr}")
```

8. Jalankan code nya di master

Dengan command mpirun np 4 -host master,slave1,slave2,slave3 python3 bubble.py