LAPORAN EKSEKUSI NUMERIK DENGAN OPEN MPI PADA UBUNTU DESKTOP

Disusun untuk memenuhi tugas Mata Kuliah Pemrosesan Paralel

Disusun Oleh :

Diki Riskiyanto	(09011282227111)
Syafaat Muhammad Bahril Huda	(09011282227075)
M. Nauval Perdana	(09011282227060)
Fakhri Naufal Dhaifullah	(09011282227108)

Dosen Pengampu :

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

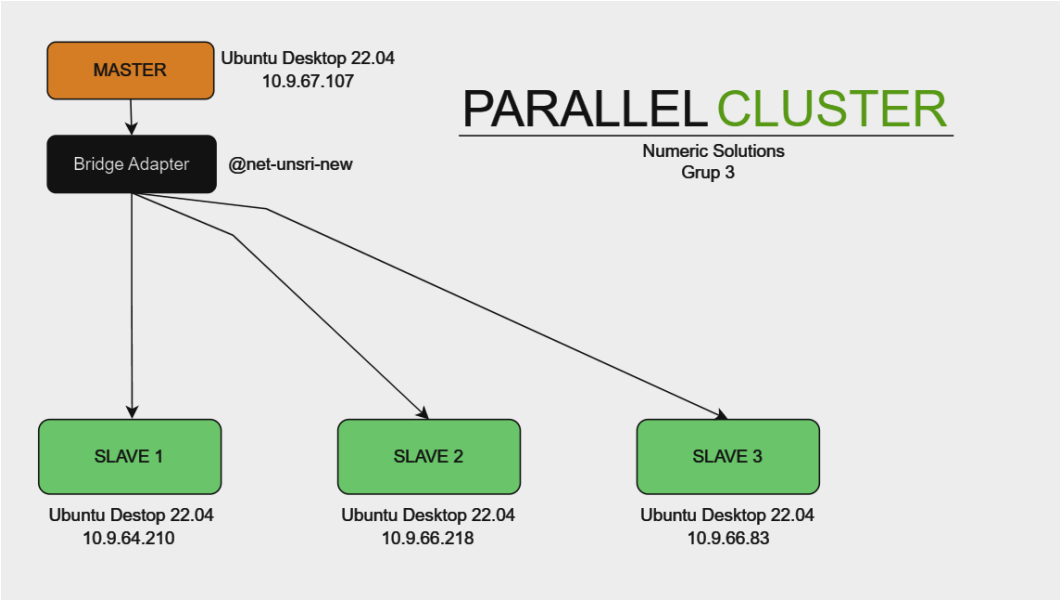
PROGRAM STUDI SISTEM KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA

2023

DEVICE DAN TOOLS YANG PERLU DISIAPKAN

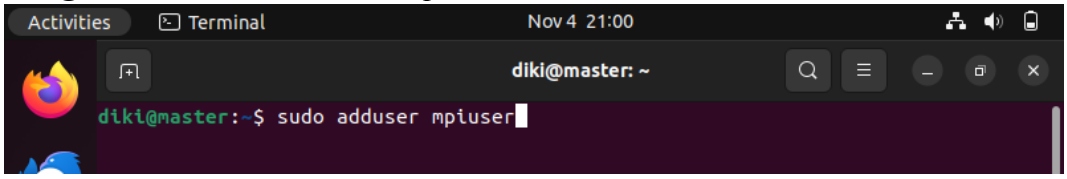
1. Ubuntu Desktop
   - Ubuntu Desktop Master
   - Ubuntu Desktop Slave 1
   - Ubuntu Desktop Slave 2
   - Ubuntu Desktop Slave 3
2. MPI (Master dan Slave)
3. SSH (Master dan Slave)
4. NFS (Master dan Slave)
5. Kodingan Numerik Python

TOPOLOGI BRIDGED



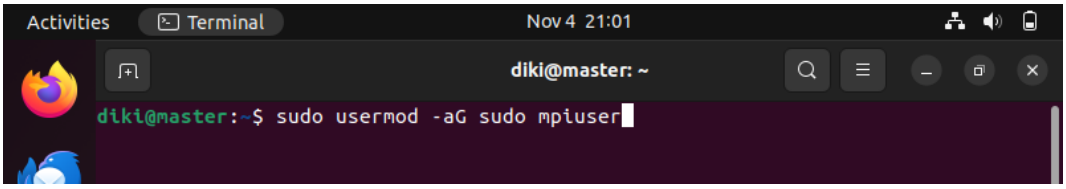
PEMBUATAN MASTER DAN SLAVE

1. Sebelum menginstal pastikan master dan setiap slave menggunakan Network Bridge Adapter, dan menggunakan internet yang sama
2. Tentukanlah device mana yang sebagai master, slave1, slave2, slave3
3. Pertama, buatlah user baru dengan perintah dibawah ini
   diki@master:~$ *sudo adduser mpiuser*



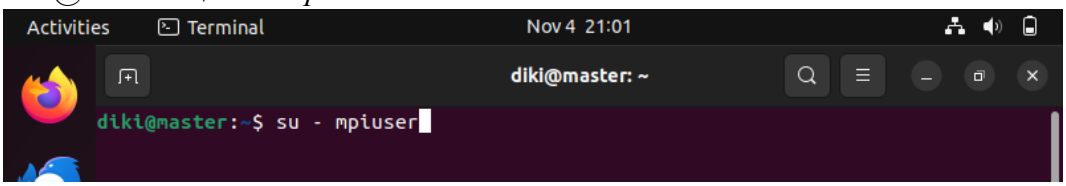
   Untuk dislave perintahnya sama, ganti bagian master menjadi slave1, slave2, dst.

4. Kemudian berikan akses kepada root dengan perintah dibawah ini
   diki@master:~$ *sudo usermod -aG sudo mpiuser*



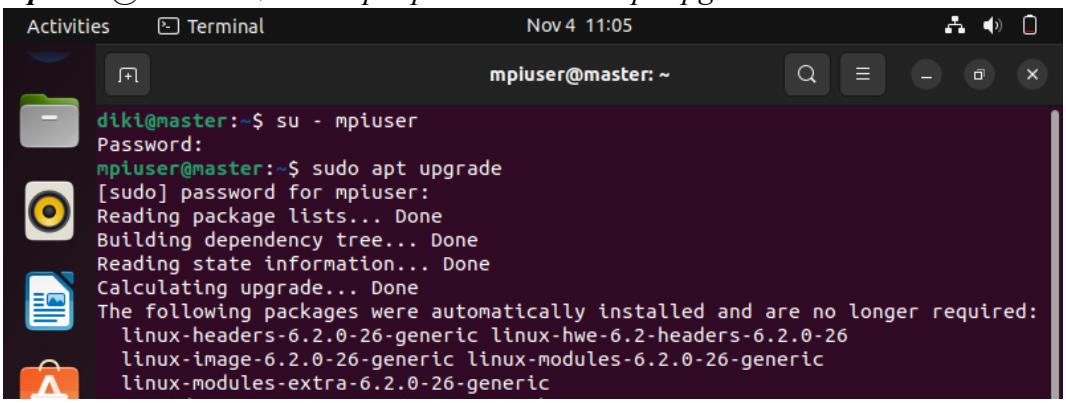
Lakukanlah perintah diatas disemua slave dengan merubah user master menjadi slave1, slave2, dst

5. Masuklah ke server dengan user dibawah ini dengan perintah berikut
   diki@master:~$ *su – mpiuser*

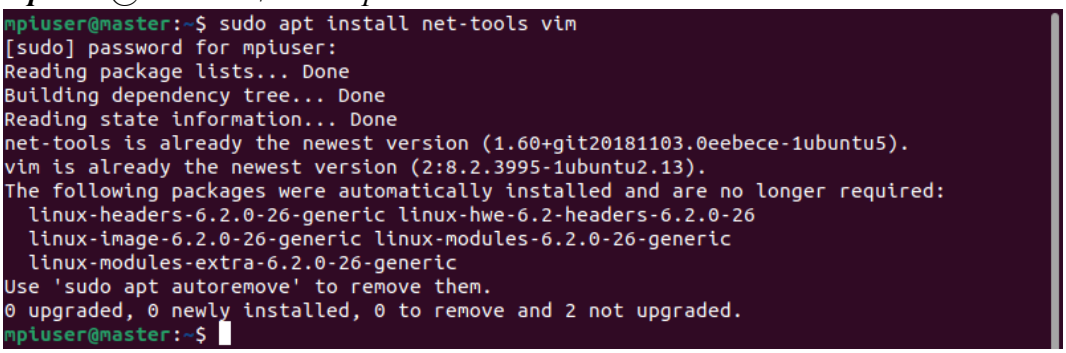


Menjadi ***mpiuser@master:~$***

6. Langkah selanjutnya update ubuntu desktop dengan perintah berikut, lalu install tools untuk mengecek Ip, vim editor teks
   mpiuser@master:~$ *sudo apt update && sudo apt upgrade*

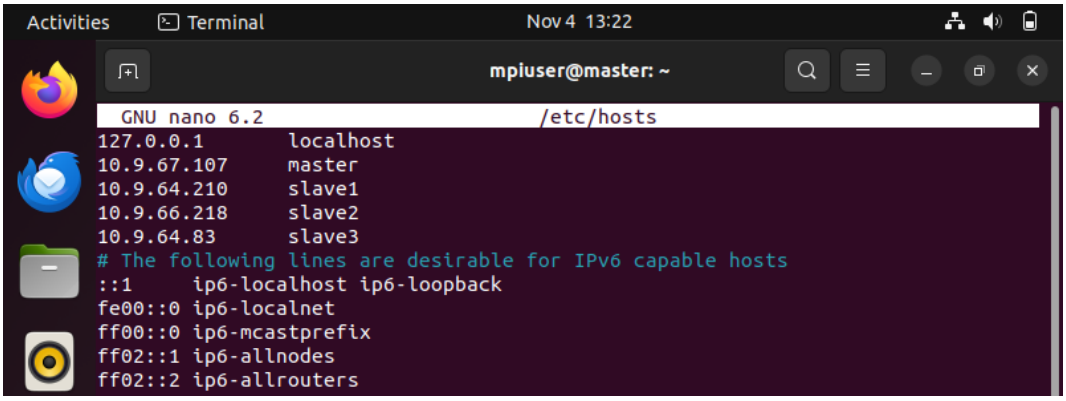


Selanjutnya install tools dengan perintah dibawah ini
mpiuser@master:~$ *sudo apt install net-tools vim*

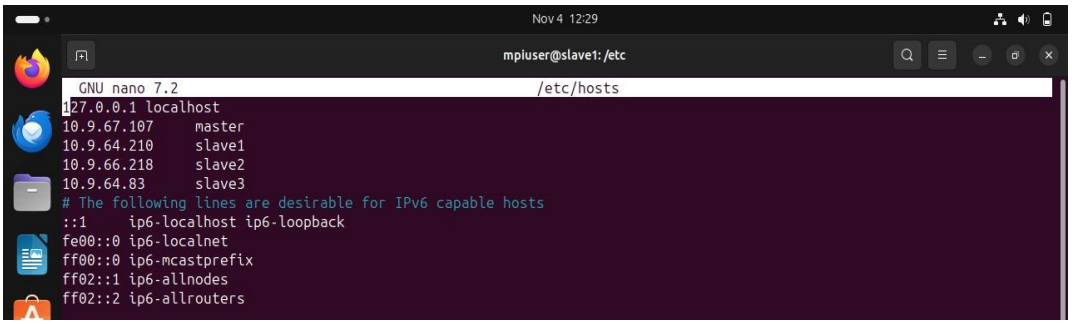


7. Selanjutnya konfigurasi file pada master, slave1, slave2, dan slave3
   mpiuser@master:~$ *cd /etc/hosts*

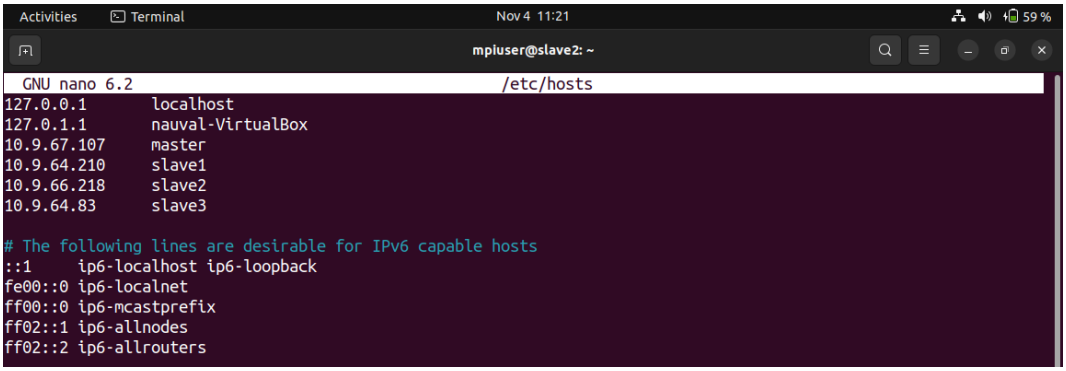
MASTER



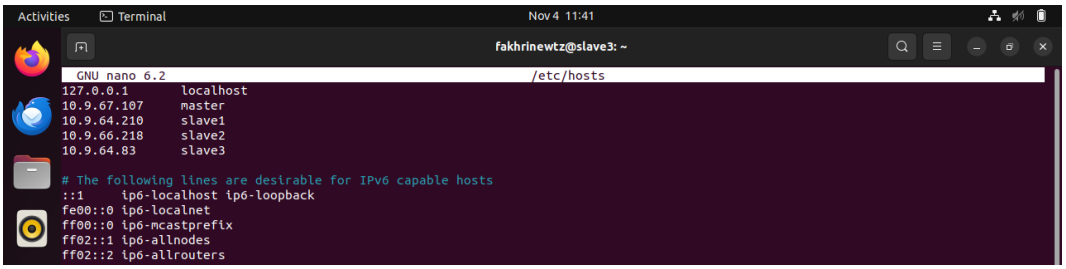
SLAVE 1



SLAVE 2



SLAVE 3



Daftarkan IP Master dan Slave berserta hostname masing masing komputer

KONFIGURASI SSH

1. Langkah berikutnya kita akan konfigurasi SSH, pertama kita install SSH. Lakukan pada master dan semua slave
   mpiuser@master:~$ *sudo apt install openssh-server*

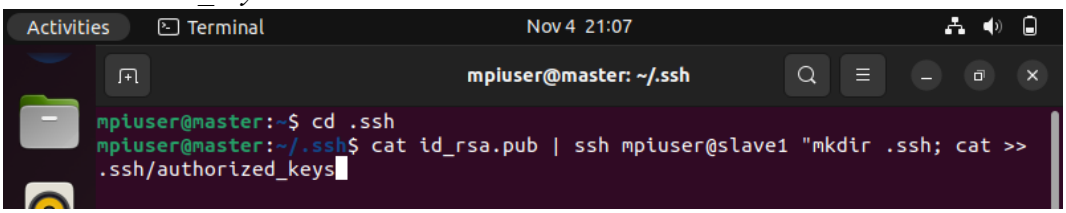
```
mpiuser@master:~$ sudo apt install openssh-server
[sudo] password for mpiuser:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:8.9p1-3ubuntu0.4).
The following packages were automatically installed and are no longer required:
  linux-headers-6.2.0-26-generic linux-hwe-6.2-headers-6.2.0-26
  linux-image-6.2.0-26-generic linux-modules-6.2.0-26-generic
  linux-modules-extra-6.2.0-26-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
mpiuser@master:~$
```

   Pastikan semua slave mengintall OpenSSH server sampai selesai dan berhasil

2. Generate key lakukan pada master saja dengan perintah berikut
   mpiuser@master:~$ *ssh-keygen -t rsa*

```
mpiuser@master:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/mpiuser/.ssh/id_rsa):
/home/mpiuser/.ssh/id_rsa already exists.
Overwrite (y/n)?
mpiuser@master:~$
```

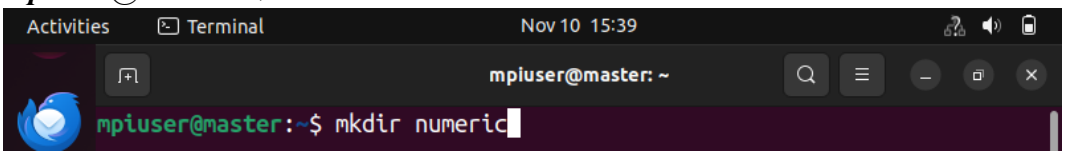
3. Copy key public ke slave (Lakukan di Master), ketikkan perintah berikut pada direktori ".ssh"
   mpiuser@master:~$ *cd .ssh*
   mpiuser@master:~/.ssh$ *cat id_rsa.pub | ssh mpiuser@slave1 "mkdir .ssh; cat >> .ssh/authorized_keys"*



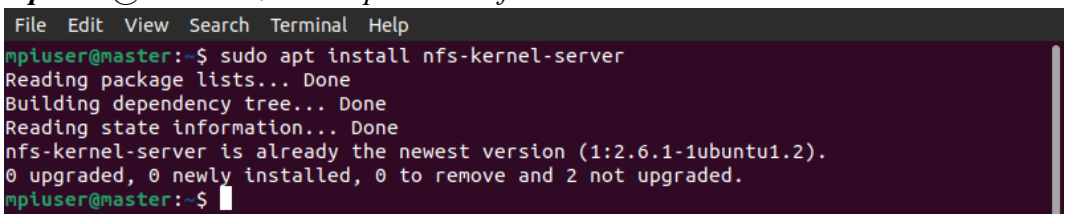
   Lakukan perintah diatas berulang kali sebanyak slave, untuk pengecekan file authorized_keys di slave, yang terletak di folder .ssh

KONFIGURASI NFS

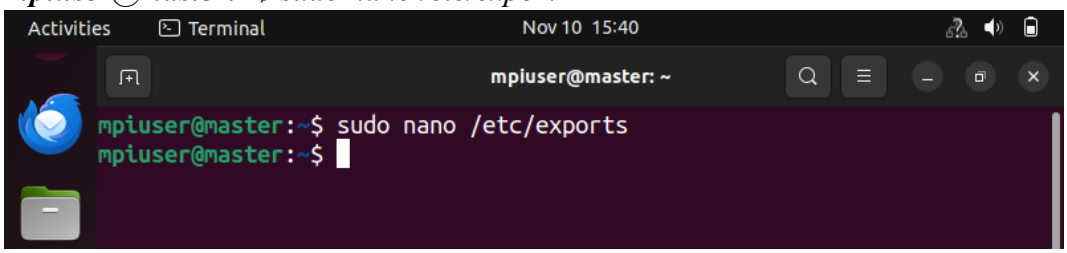
1. Buatlah shared folder, lakukanlah dimaster dan per slave
   mpiuser@master:~$ *mkdir nuumeric*



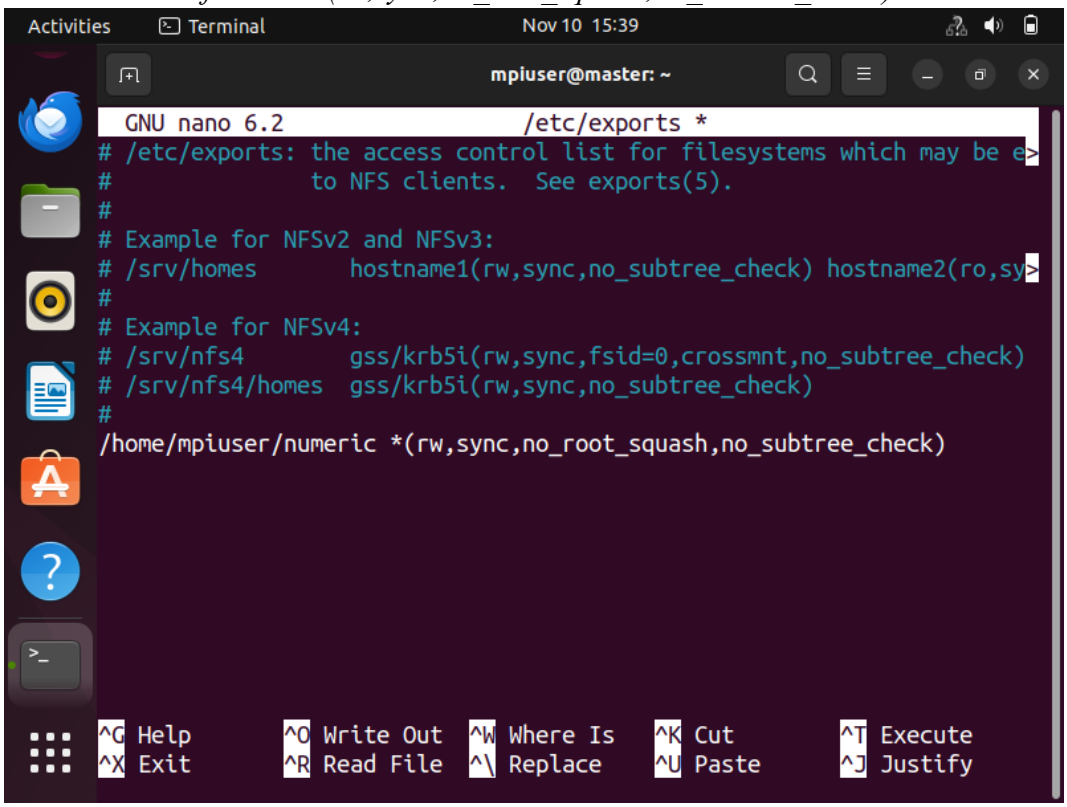
2. Selanjutnya install NFS untuk master
   mpiuser@master:~$ *sudo apt install nfs-kernel-server*



3. Lakukan konfigurasi file pada master, masuk ke file export dengan perintah
   mpiuser@master:~$ *sudo nano /etc/export*



   Konfigurasi file tambahkan commend ini pada baris berikut, ketikkan pada baris terakhir.
   *<lokasi shared folder> *(rw,sync,no_root_squash,no_subtree_check)*



   Lokasi Shared Folder merupakan tempat direktori membuat file diatas tadi

4. Kemudian ketikkan perintah berikut ini, untuk memulai kembali atau merestart NFS Server
   mpiuser@master:~$ *sudo exportfs -a*
   mpiuser@master:~$ *sudo systemctl restart nfs-kernel-server*

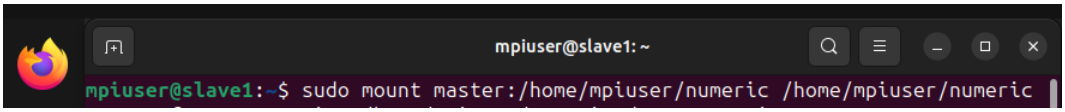
```
File  Edit  View  Search  Terminal  Help
mpiuser@master:~$ sudo exportfs -a
mpiuser@master:~$ sudo systemctl restart nfs-kernel-server
mpiuser@master:~$
```

5. Selanjutnya install NFS untuk slave
   mpiuser@slave1:~$ *sudo apt install nfs-common*

```
mpiuser@slave1:~$ sudo apt install nfs-common
[sudo] password for mpiuser:
```

   lakukan penginstalan disemua slave

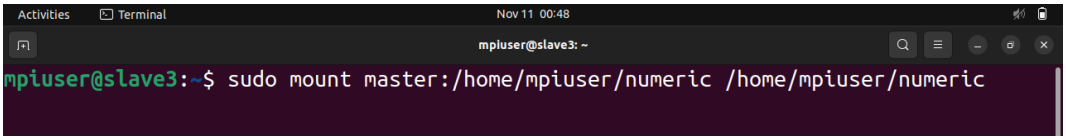
6. Setelah itu computer slave harus melakukan mounting ke computer master dengan perintah berikut
   mpiuser@slave:~$ sudo mount master:/home/mpiuser/numeric /home/mpiuser/numeric
   Slave 1



   Slave 2

```
mpiuser@slave2:~$ sudo mount master:/home/mpiuser/numeric/ /home/mpiuser/numeric/
```

   Slave 3

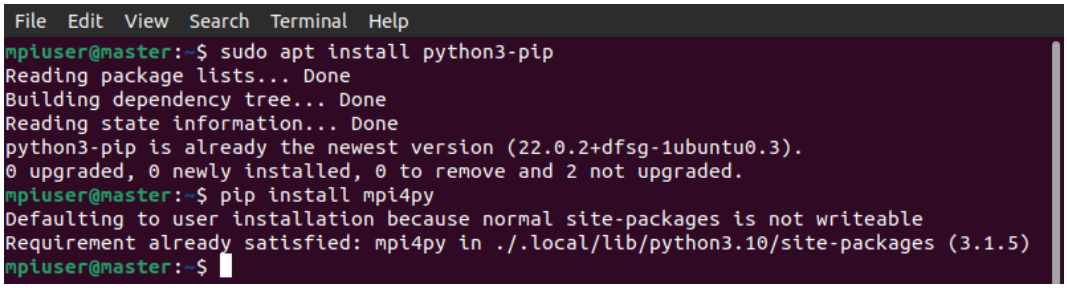


INSTALASI MPI

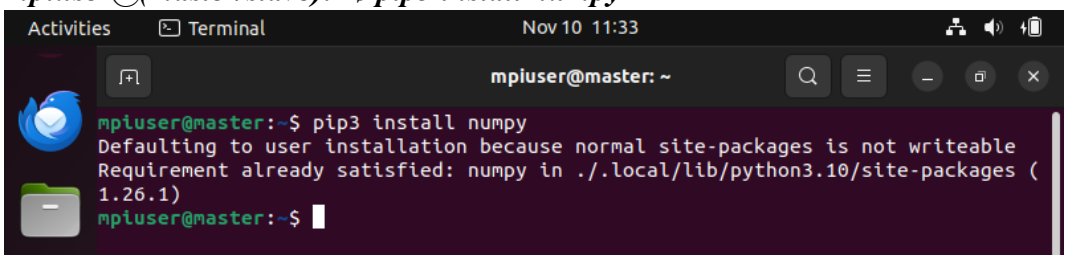
1. Instalasi MPI, lakukan pada master dan semua slave
   mpiuser@master:~$ *sudo apt install openmpi-bin libopenmpi-dev*

```
File  Edit  View  Search  Terminal  Help
mpiuser@master:~$ sudo apt install openmpi-bin libopenmpi-dev
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
libopenmpi-dev is already the newest version (4.1.2-2ubuntu1).
openmpi-bin is already the newest version (4.1.2-2ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
mpiuser@master:~$
```

2. Selanjutnya install library untuk MPI melalui pip
   mpiuser@master:~$ sudo apt install python3-pip
   mpiuser@master:~$ pip install mpi4py

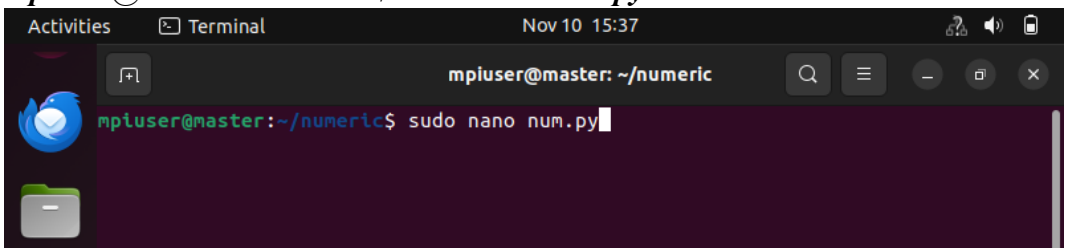


3. Lakukan penginstalan Numpy pada semua computer master dan slave guna Menyediakan Fungsi Bawaan untuk Aljabar Linier dan Pembuatan Bilangan Acak
mpiuser@(master/slave):~$ pip3 install numpy



RUNNING KODINGAN PYTHON

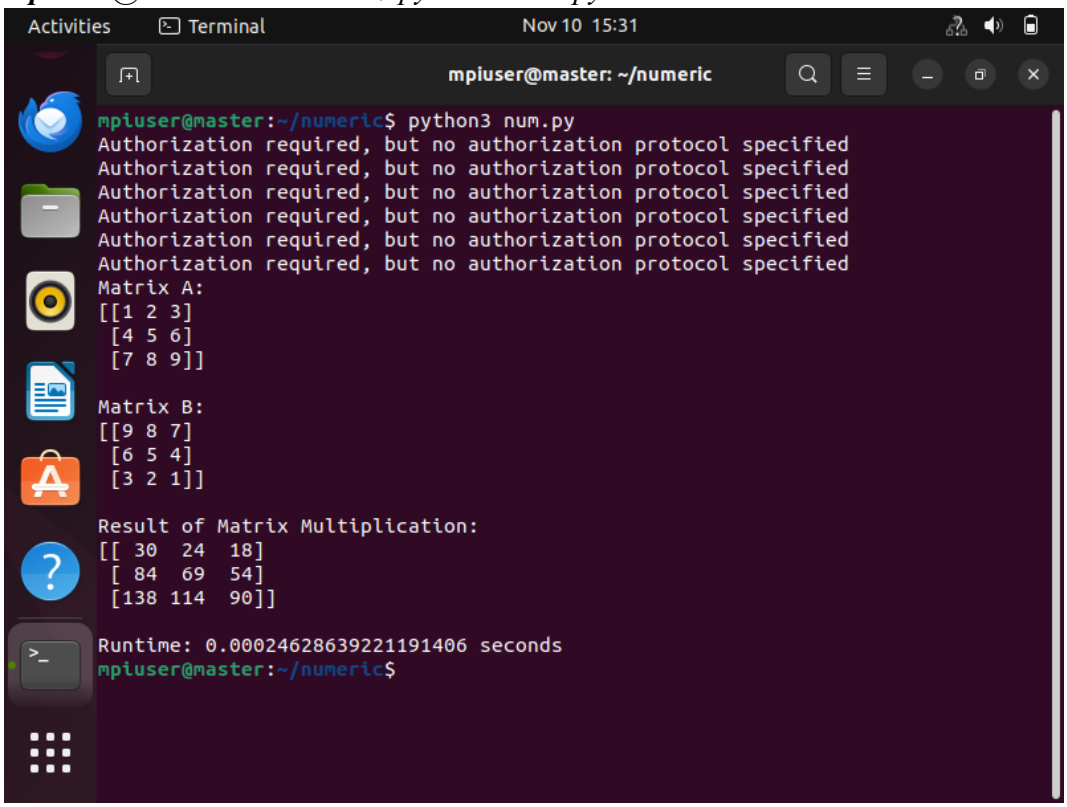
1. Bukalah direktori numeric dengan menggunakan perintah
mpiuser@master:~/numeric$ sudo nano num.py



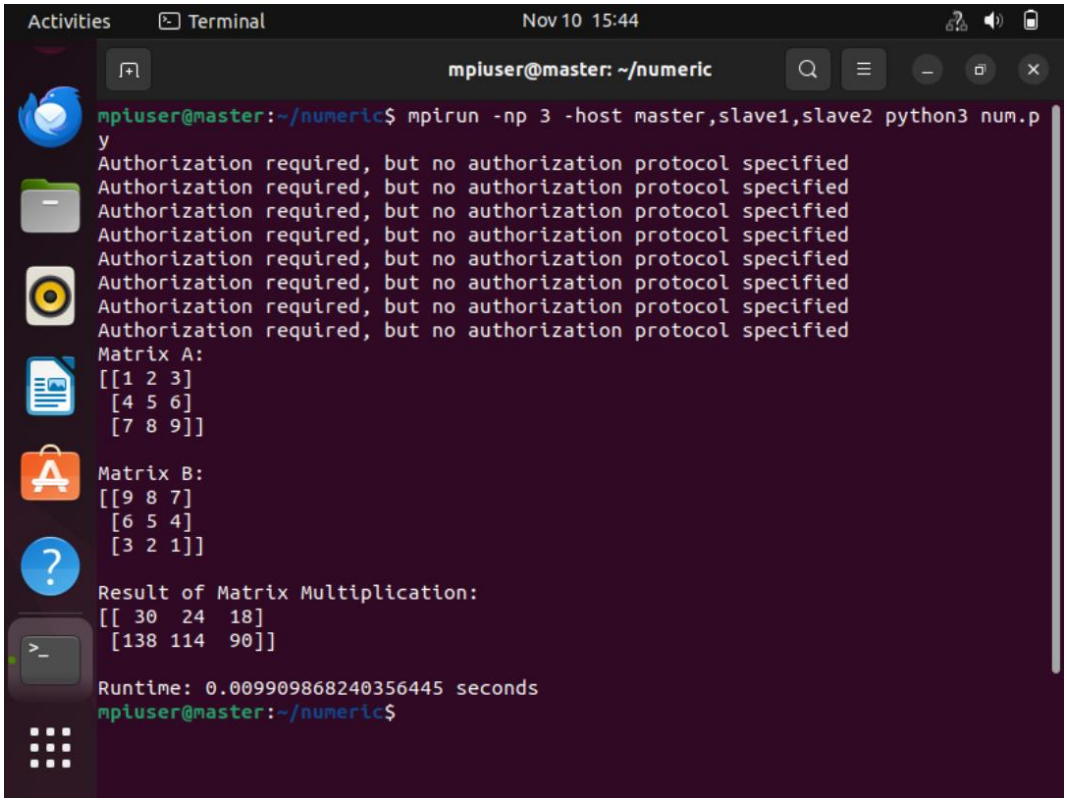
kemudian tambahkan kodingan python numerik kedalamnya

```
GNU nano 6.2                                    num.py *
from mpi4py import MPI
import numpy as np
import time

def matrix_multiply(A, B):
    return np.dot(A, B)

def scatter_data(comm, data):
    size = comm.Get_size()
    rank = comm.Get_rank()
    sendbuf = None

    if rank == 0:
        sendbuf = np.array_split(data, size)

    recvbuf = comm.scatter(sendbuf, root=0)
    return recvbuf

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

    # Define matrices A and B
    A = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
    B = np.array([[9, 8, 7], [6, 5, 4], [3, 2, 1]])

    if rank == 0:
        # Scatter matrix A to all processes
        local_A = scatter_data(comm, A)
    else:
        local_A = None

    # Broadcast matrix B to all processes
    local_B = comm.bcast(B, root=0)

    # Start the timer
    start_time = time.time()

    # Scatter matrix A and perform local matrix multiplication
    local_result = scatter_data(comm, local_A)
    local_result = matrix_multiply(local_result, local_B)

    # Gather the local results to the root process
    global_result = comm.gather(local_result, root=0)

    # Stop the timer
    end_time = time.time()

    if rank == 0:
        # Concatenate the global results to get the final result
        final_result = np.concatenate(global_result)
        print("Matrix A:")
        print(A)
        print("\nMatrix B:")
        print(B)
        print("\nResult of Matrix Multiplication:")
        print(final_result)
        print("\nRuntime: {} seconds".format(end_time - start_time))

if __name__ == "__main__":
    main()

^G Help      ^O Write Out  ^W Where Is   ^K Cut       ^T Execute   ^C Location   M-U Undo     M-A Set Mark   M-] To Bracket  ^B Previous  ^F Back
^X Exit      ^R Read File  ^\ Replace    ^U Paste     ^J Justify   ^/ Go To Line M-E Redo     M-6 Copy       ^Q Where Was    ^V Next      ^P Forward
```

2. Sebelum mengetesnya di MPI kami melakukan pengetesan langsung dengan python3 menggunakan perintah
mpiuser@master:~/numeric$ *pyhton3 num.py*



3. Setelah pengetesan python3 berhasil barulah kami eksekusi dengan mpi
mpiuser@master:~numeric$ *mpirun -np 3 -hosts master ,slave1,slave2 pyhton3 num.py*



Didapatlah hasil matriks dengan runtime selama 0.00990988240356445