

LAPORAN PRAKTIKUM 5
**“Parallel Computing menggunakan
MPI Python”**



Nama : MAYANG ARINDA TARI
NIM : 0903582226020
Prodi : Teknik Komputer
Dosen : Adi Hermansyah, S.Kom., M.T

PROGRAM STUDI TEKNIK KOMPUTER
UNIVERSITAS SRIWIJAYA
2023

PRAKTIKUM I

I. JUDUL PRAKTIKUM

Parallel Computing menggunakan MPI Python

II. TUJUAN PRAKTIKUM

1. Remote server Master dengan Putty
2. Mampu menggunakan Parallel Komputing dengan 4 Komputer sebagai ‘Slave’(penyedia resources) dan 1 Komputer sebagai Master
3. Menjalankan 1 Program pada Komputer yang berperan sebagai ‘Master’

III. ALAT

1. Virtual Box
2. Akses Internet

IV. TEORI DASAR

Parallel Computing menggunakan MPI (Message Passing Interface) adalah teknik yang memungkinkan banyak prosesor bekerja bersama untuk menyelesaikan satu masalah, dengan membagi tugas menjadi sub-tugas yang lebih kecil dan mendistribusikannya ke berbagai prosesor¹²³⁴. MPI adalah standar untuk perpustakaan pengiriman pesan untuk program paralel³. MPI-1 standar dirilis pada tahun 1994 dan standar terbaru adalah MPI-3.13.

Python, melalui modul mpi4py, menyediakan antarmuka ke MPI¹²³⁴. Modul ini didasarkan pada binding C++ MPI-2 dan hampir semua panggilan MPI didukung³. Operasi utamanya adalah metode pada objek komunikator¹²³. Modul ini mendukung komunikasi objek Python yang dapat di-pickle dan komunikasi yang dioptimalkan dari array NumPy³.

Untuk melakukan komputasi paralel, Anda tidak menggunakan beberapa thread: gunakan beberapa proses³. Modul multiprocessing memberikan API yang sangat mirip dengan modul threading yang mendukung komputasi paralel³. Ada banyak modul Python lainnya yang tersedia yang mendukung komputasi paralel³.

Program Python yang menggunakan perintah MPI harus dijalankan menggunakan interpreter MPI, yang disediakan dengan perintah mpirun³⁴. Pada beberapa sistem, perintah ini disebut mpiexec dan mpi4py tampaknya mencakup keduanya³⁴. Anda dapat menjalankan skrip Python MPI menggunakan perintah mpirun sebagai berikut: mpirun -n 4 python script.py³⁴. Di sini -n 4 memberi tahu MPI untuk menggunakan empat proses³⁴.

Dalam contoh mpi4py ini, setiap pekerja menampilkan peringkatnya dan ukuran dunia³⁴:

Kode :

```

from mpi4py import MPI
comm = MPI.COMM_WORLD
print("%d of %d" % (comm.Get_rank(), comm.Get_size()))

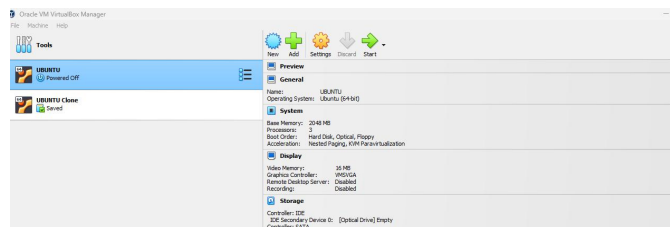
```

Anda dapat menggunakan mpirun dan python untuk menjalankan skrip ini: `mpirun -n 4 python script.py`

V. PROSEDUR PERCOBAAN

A. LOGIN KE TERMINAL UBUNTU LIVE SERVER

- Masuk ke Ubuntu lalu masukkan alamat IP kalian ke Putty, pastikan anda memilih 'ssh', dan port 22



- Setelah selesai silahkan lanjut untuk mengkonfigurasi antara master dengan Slave

```

aidil@aidil: ~
https://ubuntu.com/engage/secure-kubernetes-at-the-edge

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

58 updates can be applied immediately.
2 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

8 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Nov 21 08:41:59 2023 from 192.168.225.72
aidil@aidil:~$

```

B. Menghubungkan 1 Komputer(Master) ke 4 komputer lain(Slave)

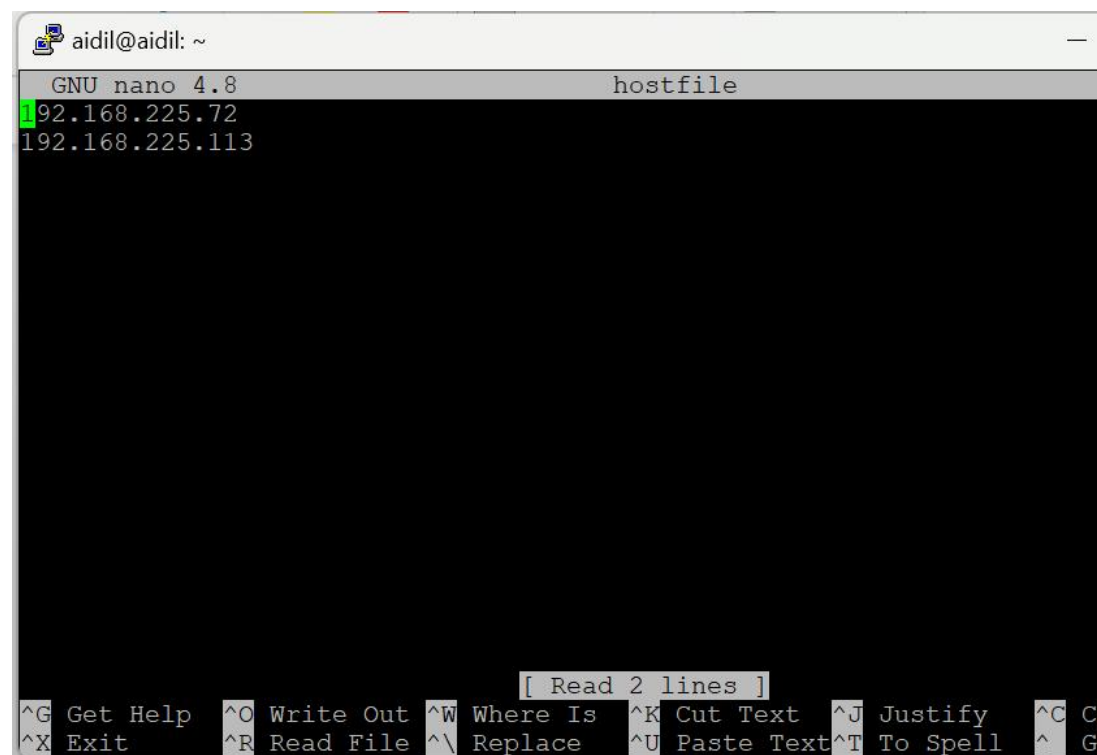
- Di setiap 'Slave' jalankan 'ssh-keygen -t

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

- Setelah itu tambahkan hasil generate kegen kamu ke 'master' dengan command '**ssh-copy-id <username>@<alamat_ip_master>**'

```
mayang@mayangarinda:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/mayang/.ssh/id_rsa):
/home/mayang/.ssh/id_rsa already exists.
Overwrite (y/n)?
mayang@mayangarinda:~$ ssh-copy-id aidil@192.168.225.243
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/mayang/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are al
eady installed
/usr/bin/ssh-copy-id: WARNING: All keys were skipped because they already exist on the remote syste
.
(if you think this is a mistake, you may want to use -f option)
```

- Selanjutnya buat kode hostfile dengan cara '**nano hostfile**'
- Masukkan IP Masing2 'Slave' kedalam nya lalu save



```
aidil@aidil: ~
GNU nano 4.8 hostfile
192.168.225.72
192.168.225.113
[ Read 2 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C C
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ G
```

C. Menjalankan 1 Program pada Komputer yang berperan sebagai 'Master'

- Jalankan satu file kode dengan '**mpirun -np <jumlah_slave> -hostfile <file_host> python cekmpi.py**'

VI. KESIMPULAN

Dalam mengatasi tantangan menyambungkan dua mesin virtual Ubuntu sebagai "Master" dan "Slave" untuk menjalankan perintah MPI, kami mulai dengan langkah-

langkah persiapan, termasuk instalasi VirtualBox, Ubuntu, dan MPI di semua mesin. Setelah itu, kami memastikan kunci SSH tanpa passphrase dihasilkan di mesin "Master" dan disalin ke "Slave", memastikan perizinan yang benar, dan memformat hostfile. Kendala muncul saat menjalankan perintah MPI, dan langkah-langkah pemecahan masalah melibatkan penyesuaian hostfile, periksa kembali kunci SSH, dan pastikan konfigurasi SSH di mesin "Slave" diatur dengan benar. Meskipun terjadi kesalahan otentikasi SSH, solusi disusun dengan cermat, termasuk perbaikan hostfile, perizinan file, dan periksa log SSH di mesin "Slave". Keseluruhan proses ini memberikan pemahaman mendalam tentang konfigurasi MPI dan manajemen otentikasi untuk menjalankan aplikasi terdistribusi secara efektif di lingkungan virtual.

Dalam perjalanan ini, penting untuk mencatat bahwa menangani kesalahan otentikasi SSH memerlukan pemahaman mendalam tentang konsep-konsep dasar MPI, VirtualBox, dan manajemen kunci SSH. Proses ini juga menekankan pentingnya format yang benar pada hostfile, dan kesiapan setiap mesin virtual untuk berkomunikasi melalui jaringan. Solusi yang telah diberikan memerlukan penggalian detail dalam konfigurasi sistem dan ketersediaan sumber daya pada setiap mesin. Meskipun dapat menantang, perjalanan ini memberikan wawasan yang berharga dalam menangani konfigurasi sistem yang kompleks dan membangun lingkungan terdistribusi yang dapat diandalkan untuk aplikasi MPI. Selanjutnya, pemahaman ini dapat diaplikasikan untuk skenario yang lebih luas dalam pengembangan aplikasi terdistribusi dan pengolahan paralel.