

**Tugas Kecil 1**  
**Pembuatan VM Linux di Cloud**  
**II2210 - Teknologi Platform**

Dipersiapkan oleh:  
**Asisten Lab Sistem Terdistribusi & Co.**



Disusun oleh :  
Nakeisha Valya Shakila  
18223133

**PROGRAM STUDI SISTEM DAN TEKNOLOGI INFORMASI**  
**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**  
**INSTITUT TEKNOLOGI BANDUNG**  
**JATINANGOR**  
**2025**

## DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>2</b>
<b>BAB I : PENDAHULUAN.....</b>	<b>3</b>
1.1. Latar Belakang Masalah.....	3
1.2. Rumusan Masalah.....	3
1.3. Tujuan dan Manfaat.....	3
<b>BAB II : PEMBAHASAN.....</b>	<b>4</b>
2.1. Installation.....	4
2.1.1. Pemilihan Cloud Provider.....	4
2.1.2. Pemilihan Distribusi Linux.....	4
2.1.3. Pembuatan dan Instalasi Virtual Machine.....	4
2.2. Post-Installation.....	6
2.2.1. Instalasi Fastfetch.....	6
2.2.2. Pembuatan User dan Group.....	6
2.2.3. Konfigurasi SSH.....	8
2.3. Web Server.....	10
2.3.1. Instalasi Python.....	10
2.3.2. Membuat Halaman dalam HTML.....	10
2.3.3. Menjalankan Web Server.....	11
<b>BAB III : PENUTUP.....</b>	<b>13</b>
3.1. Kesimpulan.....	13
<b>DAFTAR PUSTAKA.....</b>	<b>14</b>

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Masalah

Virtual Machine (VM) menjadi solusi umum dalam menjalankan sistem operasi dalam lingkungan yang terisolasi, memungkinkan pengguna untuk mengelola server, melakukan pengujian aplikasi, dan mensimulasikan jaringan tanpa mengganggu sistem utama. Dalam dunia teknologi informasi, pengembangan dan pengelolaan server web menjadi salah satu keterampilan penting, terutama bagi pengembang aplikasi dan administrator sistem. Salah satu implementasi yang sering digunakan adalah menjalankan web server sederhana menggunakan Python (`http.server`) dalam VM berbasis Ubuntu. Penggunaan Python sebagai web server memungkinkan pengguna untuk meng-host halaman web secara lokal maupun dalam lingkungan jaringan yang lebih luas.

Namun, dalam implementasinya, banyak kendala yang dapat muncul, terutama dalam konfigurasi jaringan, firewall, serta komunikasi antara sistem operasi host dan virtual machine. Salah satu permasalahan yang sering ditemui adalah server web yang berjalan di dalam VM dapat diakses menggunakan `curl` dari dalam VM itu sendiri, tetapi tidak dapat diakses melalui browser pada sistem host (misalnya Windows). Masalah ini dapat disebabkan oleh berbagai faktor, seperti pengaturan jaringan VM yang tidak sesuai (NAT, Bridged Adapter, atau Host-Only Adapter), aturan firewall (`ufw`) yang membatasi akses, port binding yang salah, atau server yang hanya mendengarkan koneksi dari `localhost`.

Untuk memastikan komunikasi yang lancar antara host dan VM, diperlukan konfigurasi yang tepat dalam beberapa aspek. Pengguna perlu memahami cara kerja jaringan dalam VM, mengatur pengikatan server ke alamat IP yang benar, serta melakukan troubleshooting jaringan agar server dapat diakses dengan lancar. Dengan pemahaman yang baik tentang konfigurasi firewall, metode binding port, dan pengaturan jaringan VM, pengguna dapat mengatasi kendala teknis dan memastikan bahwa server berjalan dengan optimal di dalam lingkungan virtual.

### 1.2. Rumusan Masalah

Dari latar belakang yang ada, terdapat beberapa rumusan masalah yang diharapkan dapat terjawab setelah membaca laporan ini, antara lain sebagai berikut:

- Bagaimana cara menjalankan server web sederhana menggunakan Python (`http.server`) di dalam Virtual Machine berbasis Ubuntu?
- Bagaimana cara mengkonfigurasi jaringan Virtual Machine agar memungkinkan akses dari host (Windows) ke server yang berjalan di dalam VM?

### 1.3. Tujuan dan Manfaat

- **Mempermudah pengguna dalam memahami konfigurasi Virtual Machine**, khususnya dalam menjalankan server web di dalamnya
- **Memastikan server web yang berjalan di dalam VM dapat diakses dengan baik**, baik dari sistem host maupun perangkat lain di jaringan

## BAB II

### PEMBAHASAN

#### 2.1. Installation

Berikut merupakan langkah-langkah untuk memilih cloud provider, menentukan distribusi Linux, dan melakukan instalasi virtual machine (VM):

##### 2.1.1. Pemilihan Cloud Provider

Cloud provider yang digunakan dalam penugasan ini adalah Microsoft Azure. Pemilihan Azure didasarkan pada kemudahan aksesnya yang tidak memerlukan kartu kredit serta adanya manfaat khusus bagi pelajar melalui program Azure for Students. Untuk keperluan tugas ini, saya menggunakan akun mahasiswa (@mahasiswa.itb.ac.id) yang terdaftar dalam program tersebut

##### 2.1.2. Pemilihan Distribusi Linux

Distribusi Linux (distro) adalah sistem operasi berbasis kernel Linux yang mencakup perangkat lunak sistem, pustaka, dan aplikasi tambahan. Setiap distro memiliki karakteristik berbeda sesuai kebutuhan. Dalam penugasan ini, distribusi Linux yang digunakan memiliki detail sebagai berikut:

- Operating System : Linux (Ubuntu Server 22.04)
- Size : Standard B1ms (1 vCPU, 2 GiB memory)
- Public IP Address : 70.153.208.25

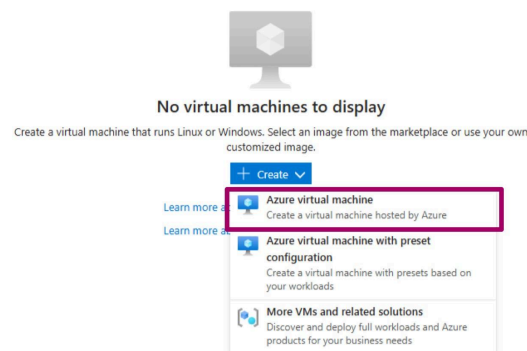
*Tabel 2.1.1 Alasan Pemilihan OS dan Size pada*

Aspek	Operating System	VM Size
Kompatibilitas	Support luas untuk docker, Python, MySQL, dll	Tersedia dalam berbagai ukuran
Biaya	Gratis (Open-Source)	Termurah di Azure dengan performa yang cukup
Skalabilitas	Mudah upgrade/downgrade versi Ubuntu	Bisa di-scale up ke ukuran lebih besar jika diperlukan.

##### 2.1.3. Pembuatan dan Instalasi Virtual Machine

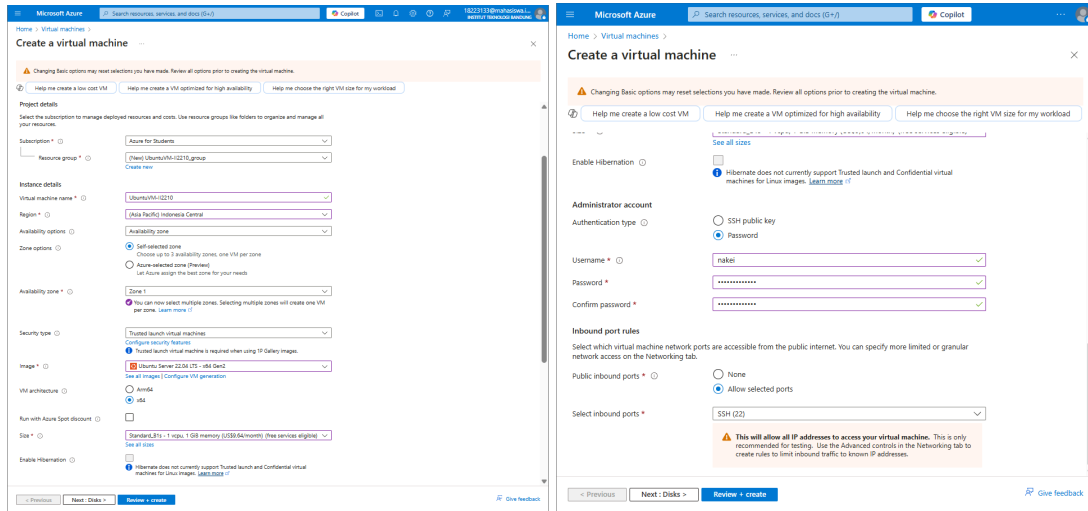
Distribusi Linux (distro) adalah sistem operasi berbasis kernel Linux yang mencakup perangkat lunak sistem, pustaka, dan aplikasi tambahan. Setiap distro memiliki karakteristik berbeda sesuai kebutuhan. Dalam penugasan ini, distribusi Linux yang digunakan memiliki detail sebagai berikut:

1. Buka dan login website Microsoft Azure
2. Klik **Create a Source > Virtual Machine > Create > Azure Virtual Machine**



*Gambar 2.1 Tampilan pengaturan dalam Microsoft Azure*

### 3. Lengkapi data virtual machine sesuai dengan kebutuhan

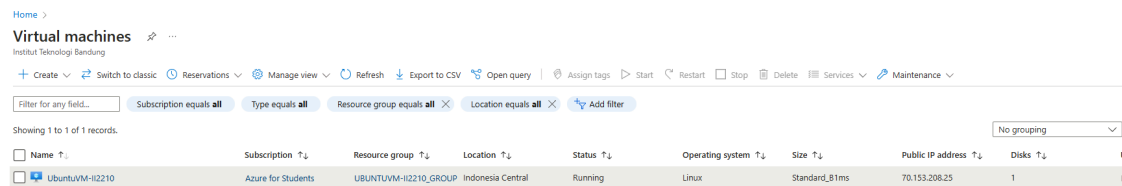


Gambar 2.2 Tampilan data yang harus dilengkapi user

Perlu diperhatikan bahwa *authentication type* yang saya gunakan berbasis password sehingga tidak ada public key hasil dari *virtual machine*-nya

### 4. Klik **Review + create** pada tombol berwarna biru di bawah halaman

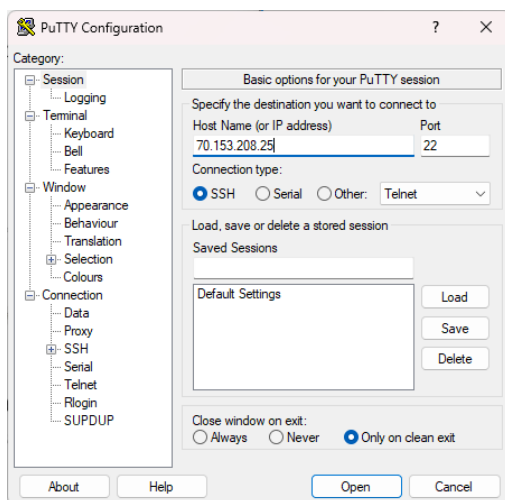
### 5. Pastikan bahwa status virtual machine dalam keadaan *running*



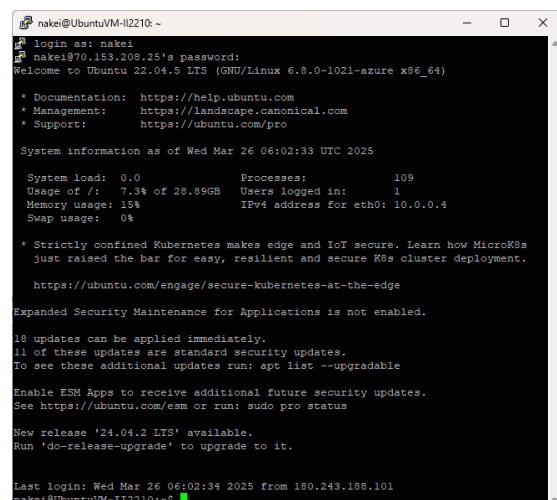
Name	Subscription	Resource group	Location	Status	Operating system	Size	Public IP address	Disks
UbuntuVM-10210	Azure for Students	UBUNTUVM-10210_GROUP	Indonesia Central	Running	Linux	Standard_B1ms	70.153.208.25	1

Gambar 2.3 Tampilan virtual machine yang berhasil dibuat

Dalam melakukan aktivitas di terminal, saya menggunakan software tambahan yaitu PuTTY, penggunaan tersebut didasari oleh terminal error pada Microsoft Azure. Untuk memastikan VM berjalan, kita bisa langsung meng-input IP Public pada PuTTY dan melakukan autentikasi/login menggunakan password yang sudah dibuat saat membuat VM



Gambar 2.4 Tampilan Software PuTTY



Gambar 2.5 Tampilan Login VM

## 2.2. Post-Installation

Setelah proses instalasi sistem operasi pada Virtual Machine selesai, langkah selanjutnya adalah melakukan konfigurasi awal agar VM siap digunakan sesuai kebutuhan. Beberapa konfigurasi penting yang perlu dilakukan meliputi **instalasi Fastfetch**, **pembuatan user dan group**, serta **konfigurasi SSH** untuk memungkinkan koneksi jarak jauh yang aman ke dalam VM.

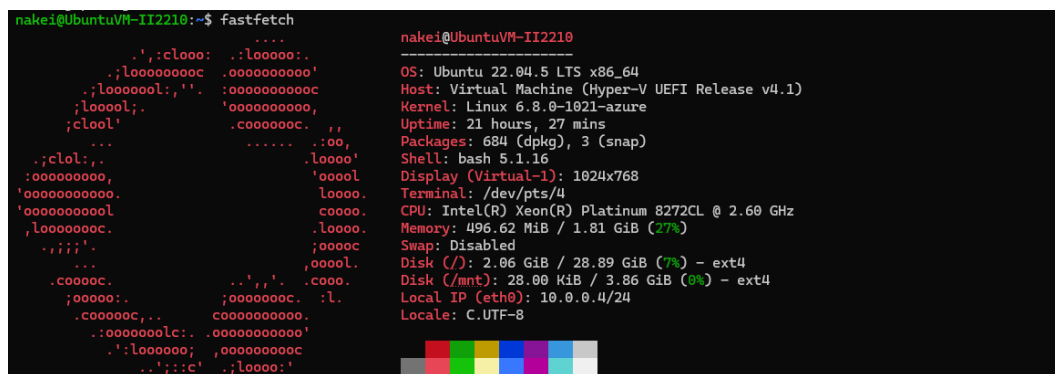
### 2.2.1. Instalasi Fastfetch

**Fastfetch** merupakan program ringan yang digunakan untuk menampilkan informasi sistem di terminal, seperti spesifikasi hardware, sistem operasi, dan versi kernel. Berikut adalah langkah-langkah untuk menginstal dan menjalankan Fastfetch di terminal **Ubuntu Server 22.04** :

```
nakei@UbuntuVM-II2210:~$ sudo add-apt-repository  
ppa:zhangsongcui3371/fastfetch
```

```
nakei@UbuntuVM-II2210:~$ sudo apt install fastfetch
```

```
nakei@UbuntuVM-II2210:~$ fastfetch
```



Gambar 2.6 Tampilan Instalasi Fastfetch Berhasil

### 2.2.2. Pembuatan User dan Group

Dalam sistem operasi berbasis Linux, **user (pengguna)** dan **group (grup)** digunakan untuk mengelola akses dan izin dalam sistem, termasuk dalam lingkungan **Virtual Machine (VM)**.

- **User** adalah akun yang digunakan untuk masuk ke sistem dan menjalankan perintah. Setiap user memiliki direktori home sendiri untuk menyimpan file dan konfigurasi pribadi.
- **Group** adalah kumpulan user yang memiliki hak akses yang sama terhadap file dan sumber daya tertentu, sehingga memudahkan pengelolaan izin dalam sistem.

Dalam sistem operasi berbasis Linux, **user (pengguna)** dan **group (grup)** digunakan untuk mengelola akses dan izin dalam sistem, termasuk dalam lingkungan **Virtual Machine (VM)**. Berikut merupakan langkah-langkah untuk membuat group :

```
nakei@UbuntuVM-II2210:~$ sudo addgroup <nama_group>
```

```
nakei@UbuntuVM-II2210:~$ sudo usermod -aG <nama_group>  
<nama_user>
```

Apabila group sudah berhasil dibuat, kemudian kita bisa langsung membuat user dan melakukan klasifikasi group pada masing masing user, berikut langkah langkahnya

```
nakei@UbuntuVM-II2210:~$ sudo adduser <nama_user>
```

```
nakei@UbuntuVM-II2210:~$ sudo usermod -aG sudo <nama_user>
```

Untuk memastikan group sudah diklasifikasikan berdasarkan user, kita bisa langsung mengeceknya dengan command dibawah ini

```
nakei@UbuntuVM-II2210:~$ groups <nama_user>
```

```
nakei@UbuntuVM-II2210:~$ groups vaelya
vaelya : vaelya sudo administrator
nakei@UbuntuVM-II2210:~$ groups sister
sister : sister sudo administrator
nakei@UbuntuVM-II2210:~$ groups raidenei
raidenei : raidenei waifu
```

Gambar 2.7 Tampilan group dan user sesuai spesifikasi soal

```
nakei@UbuntuVM-II2210:~$ getent passwd vaelya
vaelya:x:1002:1004:::/home/vaelya:/bin/bash
nakei@UbuntuVM-II2210:~$ getent passwd sister
sister:x:1001:1002:::/home/sister:/bin/bash
nakei@UbuntuVM-II2210:~$ getent passwd raidenei
raidenei:x:1003:1005:::/home/raidenei:/bin/bash
```

Gambar 2.8 Tampilan setiap user memiliki direktori home

Perlu diketahui bahwa group **sudo** merupakan default sehingga tidak perlu lagi untuk dibuat, berbeda dengan **administrator** dan **waifu** yang perlu dibuat berdasarkan cara diatas. Dalam Linux, pengelolaan izin pada direktori home penting untuk keamanan dan keteraturan sistem. Dengan izin yang tepat, pengguna hanya dapat mengakses file mereka sendiri, mencegah akses atau modifikasi yang tidak valid. Berikut langkah-langkah mengelola izin pada grup yang telah dibuat:

Tabel 2.1.2 Hak Akses File dalam Sistem UNIX/Linux

Octal	Binary	File Mode
0	000	---
1	001	--x
2	010	-w-
3	011	-wx
4	100	r--
5	101	r-x
6	110	rw-
7	111	rwx

Pertama-tama kita perlu mengetahui mode izin file dalam format oktal pada sistem operasi berbasis Linux. Setiap angka mewakili izin yang diberikan kepada **user**, **group**, dan **others**. Setiap izin memiliki nilai sebagai berikut:

- Read (r) = 4
- Write (w) = 2
- Execute (x) = 1

Untuk mengubah izin akses direktori, kita perlu memindahkan kepemilikan ke user yang benar dengan menuliskan command dibawah ini

```
nakei@UbuntuVM-II2210:~$ sudo chown <nama_user>: <grup_user> /home/<nama_user>
```

Setelah itu, kita bisa mengatur izin aksesnya dengan menuliskan command dibawah ini

```
nakei@UbuntuVM-II2210:~$ sudo chmod <izin> /home/<nama_user>
```

Perlu diperhatikan bahwa group administrator berada pada dua user yaitu **vaelya** (user pribadi) dan user **sister**, sehingga pada perizinannya perlu dituliskan pada masing masing user

```
nakei@UbuntuVM-II2210:~$ sudo chown administrator:administrator /home/administrator
nakei@UbuntuVM-II2210:~$ sudo chown vaelya:administrator /home/vaelya
nakei@UbuntuVM-II2210:~$ sudo chown raidenei:waifu /home/raidenei
nakei@UbuntuVM-II2210:~$ sudo chmod 750 /home/administrator
nakei@UbuntuVM-II2210:~$ sudo chmod 750 /home/vaelya
nakei@UbuntuVM-II2210:~$ sudo chmod 755 /home/raidenei
```

Gambar 2.9 Tampilan izin akses direktori berdasarkan spesifikasi soal

```
nakei@UbuntuVM-II2210:~$ sudo chmod 750 /home/administrator
nakei@UbuntuVM-II2210:~$ sudo chmod 750 /home/vaelya
nakei@UbuntuVM-II2210:~$ sudo chmod 755 /home/raidenei
```

Gambar 2.10 Tampilan setiap user memiliki direktori home

Untuk memastikan permission berhasil diubah, kita bisa langsung mengeceknya dengan command dibawah ini

```
nakei@UbuntuVM-II2210:~$ ls -ld /home/<nama_user1>
/home/<nama_user2> /home/<nama_user3>
```

```
nakei@UbuntuVM-II2210:~$ ls -ld /home/vaelya /home/raidenei /home/sister
drwxr-xr-x 2 raidenei waifu 4096 Mar 25 02:59 /home/raidenei
drwxr-xr-x 2 sister administrator 4096 Mar 24 17:41 /home/sister
drwxr-xr-x 2 vaelya administrator 4096 Mar 25 02:57 /home/vaelya
```

Gambar 2.11 Tampilan setiap user memiliki direktori home

### 2.2.3. Konfigurasi SSH

Dalam sistem operasi berbasis Linux, **user (pengguna)** dan **group (grup)** digunakan untuk mengelola akses dan izin dalam sistem, termasuk dalam lingkungan **Virtual Machine (VM)**.

```
nakei@UbuntuVM-II2210:~$ ssh-keygen -t rsa -b 4096 -C <Komentar>
```

```
nakei@UbuntuVM-II2210:~$ ssh-keygen -t rsa -b 4096 -C "vaelya@UbuntuVM-II2210"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/nakei/.ssh/id_rsa):
/home/nakei/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/nakei/.ssh/id_rsa
Your public key has been saved in /home/nakei/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:qrflckG4llUbuJGlcQvJhTUMyU0yc/O3uA8BAN82J0 vaelya@UbuntuVM-II2210
The key's randomart image is:
+----[RSA 4096]-----+
| . B#t= |
| +. @E. |
| .o++ |
| . oooo |
|... . = S |
|.o =. |
| = =.o |
|.o..* |
|.. ooo.o |
+----[SHA256]-----+
```

Gambar 2.12 Tampilan kunci SSH berhasil dibuat

Untuk memastikan bahwa kunci SSH berhasil dibuat untuk diri sendiri, kita bisa langsung mengeceknya dengan command dibawah ini

```
nakei@UbuntuVM-II2210:~$ ls -l ~/.ssh/id_rsa*
```

```
nakei@UbuntuVM-II2210:~$ ls -l ~/.ssh/id_rsa*
-rw----- 1 nakei nakei 3389 Mar 25 18:16 /home/nakei/.ssh/id_rsa
-rw-r--r-- 1 nakei nakei 748 Mar 25 18:16 /home/nakei/.ssh/id_rsa.pub
```

Gambar 2.13 Tampilan verifikasi bahwa kunci SSH berhasil dibuat

Untuk menambahkan kunci SSH ke user yang sudah ditambahkan sebelumnya (**vaelya**), kita bisa menyalin kunci publik dengan menggunakan command dibawah ini

```
nakei@UbuntuVM-II2210:~$ ssh-copy-id <nama_user>@localhost
```

```
nakei@UbuntuVM-II2210:~$ ssh-copy-id vaelya@localhost
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/nakei/.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 already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
vaelya@localhost's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'vaelya@localhost'"
and check to make sure that only the key(s) you wanted were added.
```

Gambar 2.14 Tampilan kunci SSH yang berhasil ditambahkan

Apabila konfigurasi sudah berhasil, kita bisa melakukan uji coba login ke user yang telah dibuat (**vaelya**) menggunakan SSH tanpa password



```
nakei@UbuntuVM-II2210:~$ ssh <nama_user>@localhost
```

Dalam meningkatkan keamanan pada sistem operasi, kita akan **memblokir akses SSH untuk user root** serta **menonaktifkan login berbasis password**, sehingga hanya pengguna dengan kunci SSH yang dapat masuk. Untuk mematikan login root via SSH dan dengan password dapat dilakukan dengan mengedit konfigurasi SSH

```
nakei@UbuntuVM-II2210:~$ sudo nano /etc/ssh/sshd_config
```

Apabila telah dilakukan perubahan, selanjutnya perlu dilakukan restart terhadap layanan SSH

```
nakei@UbuntuVM-II2210:~$ sudo systemctl restart ssh
```

Untuk memastikan akses SSH untuk user root diblokir dan login berbasis password dimatikan, maka bisa menggunakan command dibawah

```
nakei@UbuntuVM-II2210:~$ sudo cat /etc/ssh/sshd_config | grep PasswordAuthentication
```

```
nakei@UbuntuVM-II2210:~$ sudo cat /etc/ssh/sshd_config | grep PermitRootLogin
```

```
nakei@UbuntuVM-II2210:~$ sudo cat /etc/ssh/sshd_config | grep PasswordAuthentication
PasswordAuthentication no
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication, then enable this but set PasswordAuthentication
nakei@UbuntuVM-II2210:~$ sudo cat /etc/ssh/sshd_config | grep PermitRootLogin
PermitRootLogin no
# the setting of "PermitRootLogin without-password".
```

Gambar 2.15 Tampilan akses telah diblokir dan dimatikan

Apabila berhasil memblokir login dengan password, maka kita bisa mengeceknya dengan command dibawah ini

```
nakei@UbuntuVM-II2210:~$ ssh -o PubkeyAuthentication=no <nama_user>@localhost
```

```
nakei@UbuntuVM-II2210:~$ ssh -o PubkeyAuthentication=no vaelya@localhost
vaelya@localhost: Permission denied (publickey).
```

Gambar 2.16 Tampilan SSH tidak memiliki izin untuk akses

Autentikasi berbasis SSH key merupakan metode aman untuk mengakses server tanpa menggunakan kata sandi. Berdasarkan spesifikasi soal, diharuskan menambahkan kunci SSH bagi user **sister** dan **raidenei**, memastikan akses yang lebih aman dan efisien. Berikut merupakan langkah untuk menambahkan public key pada masing masing user :

```
# membuat folder .ssh di dalam direktori home
nakei@UbuntuVM-II2210:~$ sudo mkdir -p /home/<nama_user>/.ssh
```

```
#menyimpan public key
nakei@UbuntuVM-II2210:~$ sudo touch
/home/<nama_user>/.ssh/authorized_keys
```

Kemudian, kita bisa langsung menambahkan public key ke authorized\_keys berdasarkan spesifikasi soal

```
nakei@UbuntuVM-II2210:~$ echo "public-key" | sudo tee -a
/home/<nama_user>/.ssh/authorized_keys
```

Atur izin file dan direktori sesuai dengan keamanan SSH

```
nakei@UbuntuVM-II2210:~$ sudo chmod 700 /home/<nama_user>/.ssh
```

```
nakei@UbuntuVM-II2210:~$ sudo chmod 600
/home/<nama_user>/.ssh/authorized_keys

# Lakukan verifikasi bahwa kunci telah ditambahkan
nakei@UbuntuVM-II2210:~$ sudo cat
/home/<nama_user>/.ssh/authorized_keys

# Lakukan restart agar perubahan dapat diterapkan
nakei@UbuntuVM-II2210:~$ sudo systemctl restart ssh

nakei@UbuntuVM-II2210:~$ sudo cat /home/sister/.ssh/authorized_keys
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIMu0vMDET4Pms2/8mpOuDHV3yCTJX7oMUz6+TuNp0g05 sister
nakei@UbuntuVM-II2210:~$ sudo cat /home/raidenei/.ssh/authorized_keys
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAICcYP1o5yiPhhoVuREcR00FXq76sZr4lkCojLq6pjmOf raidenei
```

Gambar 2.17 Tampilan public key yang berhasil ditambahkan

## 2.3. Web Server

Pada tahap ini, memastikan sistem siap untuk menjalankan web server sederhana, memungkinkan pengguna menghosting halaman web secara lokal. Dimulai dengan instalasi Python sebagai interpreter utama, kemudian pembuatan file **index.html** sebagai halaman utama yang akan ditampilkan, hingga menjalankan web server pada port 7787 dengan konfigurasi agar tetap aktif meskipun sesi SSH ditutup. Berikut adalah langkah-langkah yang perlu dilakukan :

### 2.3.1. Instalasi Python

Lakukan instalasi python pada command virtual machine

```
#update daftar paket sistem
nakei@UbuntuVM-II2210:~$ sudo apt update

#install Python3 dan pip
nakei@UbuntuVM-II2210:~$ sudo apt install python3 python3-pip -y

#verifikasi instalasi Phyton dan Pip
nakei@UbuntuVM-II2210:~$ python3 --version
nakei@UbuntuVM-II2210:~$ pip3 --version

nakei@UbuntuVM-II2210:~$ python3 --version
Python 3.10.12
nakei@UbuntuVM-II2210:~$ pip3 --version
pip 22.0.2 from /usr/lib/python3/dist-packages/pip (python 3.10)
```

Gambar 2.18 Tampilan instalasi Python berhasil

### 2.3.2. Membuat Halaman dalam HTML

Pembuatan halaman **index.html** dan menyimpannya di direktori home bertujuan untuk menyediakan halaman utama yang dapat diakses melalui web server

```
#membuka text editor
nakei@UbuntuVM-II2210:~$ nano ~/index.html
```

Setelah membuka text-editor, vm akan menampilkan plain text, hal tersebut memungkinkan untuk kita menyimpan code index.html

```
code index.html
```

Pastikan bahwa file telah ter-save dengan mengklik **CTRL+X > Y > Enter**. Kemudian simpan file tersebut di direktori home

```
nakei@UbuntuVM-II2210:~$ ls ~/index.html
```

```
/home/<nama_user_main>/index.html
```

### 2.3.3. Menjalankan Web Server

Web server diperlukan untuk **menyediakan akses ke file dalam direktori tertentu melalui protokol HTTP**, sehingga halaman web seperti **index.html** dapat diakses dari browser.

```
#membuka text editor
nakei@UbuntuVM-II2210:~$ nano ~/webserver.py
```

Setelah membuka text-editor, vm akan menampilkan plain text, hal tersebut memungkinkan untuk kita menyimpan code `webserver.py`

```
import http.server
import socketserver

PORT = 7787
DIRECTORY = "/home/nakei"

class MyHandler(http.server.SimpleHTTPRequestHandler):
    def __init__(self, *args, **kwargs):
        super().__init__(*args, directory=DIRECTORY, **kwargs)

def run_server():
    with socketserver.TCPServer(("", PORT), MyHandler) as httpd:
        print(f"Server berjalan di port {PORT}")
        httpd.serve_forever()

if __name__ == "__main__":
    run_server()
```

Pastikan bahwa file telah ter-save dengan mengklik **CTRL+X > Y > Enter**. Kemudian simpan file tersebut di direktori home

```
nakei@UbuntuVM-II2210:~$ chmod +x ~/webserver.py
```

Apabila server sudah siap dan `index.html` sudah dibuat, maka kita harus menyalakan server selama akses web berjalan

```
nakei@UbuntuVM-II2210:~$ python3 ~/webserver.py
```

Berdasarkan percobaan saya dalam menyusun web server, terkadang terdapat kendala yang sebenarnya apabila ditinjau dari step sebelumnya tidak ada yang salah karena ketika dicek pada VM file `index.html` masih berjalan tetapi saat di local web tidak berjalan, maka terdapat alternatif sebagai berikut

```
#melihat informasi koneksi dan port yang sedang aktif
nakei@UbuntuVM-II2210:~$ sudo apt install net-tools
```

```
#memaksa menutup proses yang menggunakan port 7787
nakei@UbuntuVM-II2210:~$ sudo fuser -k 7787/tcp
```

```
#menjalankan web
nakei@UbuntuVM-II2210:~$ python3 -/webserver.py
```

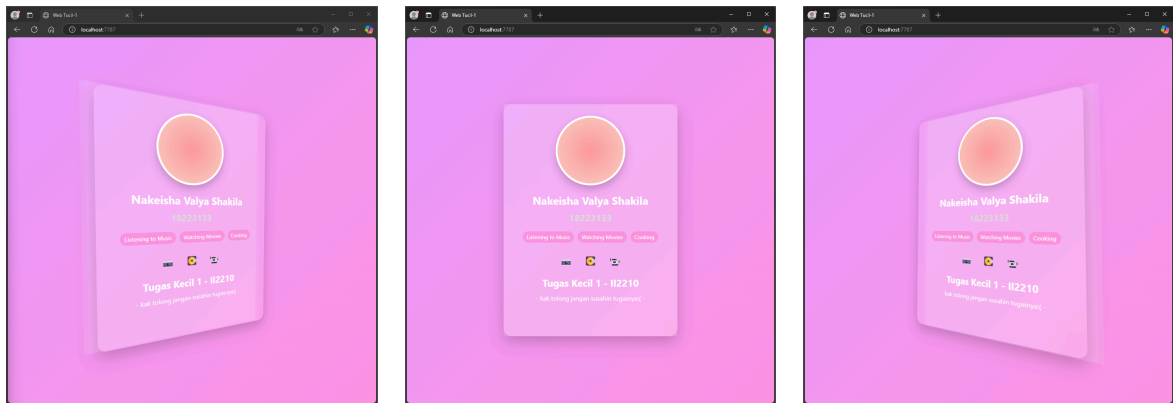
Untuk mengakses tampilan `index.html` bisa dilakukan dengan mengecek di browser (`http://localhost:7787`) dan bisa juga menggunakan command dibawah untuk menguji apakah server web berfungsi tanpa harus menggunakan browser

```
nakei@UbuntuVM-II2210:~$ curl http://localhost:7787/index.html
```

```
nakei@UbuntuVM-II2210:~$ # Lihat konten file
cat ~/index.html
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <title>Web Tugil-1</title>
  <style>
    * {
      margin: 0;
      padding: 0;
      box-sizing: border-box;
    }
    body {
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
      background: linear-gradient(135deg, #ea98ff, #ff95e4);
      min-height: 100vh;
      display: flex;
      justify-content: center;
      align-items: center;
      perspective: 1000px;
      overflow: hidden;
      color: white;
    }
    .container {
      position: relative;
      width: 450px;
      height: 600px;
      background: rgba(255,255,255,0.1);
      border-radius: 20px;
      backdrop-filter: blur(15px);
      display: flex;
      flex-direction: column;
      align-items: center;
      padding: 30px;
      transform-style: preserve-3d;
      transition: all 0.3s ease;
      box-shadow: 0 15px 35px rgba(0,0,0,0.2);
    }
    .layer {
      position: absolute;

```

Gambar 2.19 Tampilan file index.html dengan server sudah berjalan



Gambar 2.20 Tampilan website interaktif terhadap cursor

```
nakei@UbuntuVM-II2210:~$ python3 webserver.py
Server berjalan di port 7787
127.0.0.1 - - [26/Mar/2025 00:10:32] "GET / HTTP/1.1" 304 -
127.0.0.1 - - [26/Mar/2025 00:10:36] "GET / HTTP/1.1" 304 -
```

Gambar 2.21 Server berjalan

## **BAB III**

### **PENUTUP**

#### **3.1. Kesimpulan**

Virtual Machine (VM) adalah solusi yang banyak digunakan dalam pengelolaan server dan pengujian aplikasi dalam lingkungan yang terisolasi. Dalam konteks ini, menjalankan server web menggunakan Python (`http.server`) dalam VM berbasis Ubuntu menjadi salah satu implementasi yang sering digunakan. Namun, terdapat beberapa kendala teknis, seperti konfigurasi jaringan yang tidak sesuai, pengaturan firewall yang membatasi akses, serta masalah port binding yang dapat menghambat konektivitas antara host dan VM.

Untuk mengatasi permasalahan tersebut, pengguna perlu memahami konfigurasi jaringan VM, memastikan server web dapat diakses dari host, serta melakukan troubleshooting jaringan yang mencakup pemahaman mengenai metode binding port dan pengaturan firewall. Implementasi VM dilakukan menggunakan Microsoft Azure karena kemudahan aksesnya, terutama bagi mahasiswa yang terdaftar dalam program Azure for Students. Proses instalasi VM melibatkan pemilihan distribusi Linux (Ubuntu Server 22.04), pembuatan VM melalui Azure, serta konfigurasi awal menggunakan PuTTY sebagai terminal akses.

Setelah VM berhasil dibuat, beberapa konfigurasi lanjutan dilakukan, seperti instalasi `Fastfetch` untuk menampilkan informasi sistem, pembuatan user dan group untuk manajemen akses, serta pengaturan izin direktori home agar sesuai dengan standar keamanan. Selain itu, konfigurasi Secure Shell (SSH) juga diterapkan untuk memungkinkan koneksi jarak jauh yang aman ke dalam VM. Keseluruhan proses ini bertujuan untuk memastikan server web dalam VM dapat berjalan dengan optimal serta dapat diakses dari host maupun perangkat lain dalam jaringan.

## DAFTAR PUSTAKA

- Erik Dubois. (2024, September 28). Arco : 4271 How to add fastfetch to Ubuntu - workflow video - search for the source [Video]. YouTube.  
[https://www.youtube.com/watch?v=Rb8yt\\_HCmwY](https://www.youtube.com/watch?v=Rb8yt_HCmwY)
- A Computer Guru. (2023, December 16). How to create a Linux VM in Azure Portal | Deploy Ubuntu Virtual Machine in Azure - step by step [Video]. YouTube.  
<https://www.youtube.com/watch?v=RgNFd2AM7hs>
- Change Azure VM authentication to ssh key. (n.d.). Stack Overflow.  
<https://stackoverflow.com/questions/74871601/change-azure-vm-authentication-to-ssh-key>
- Tagliaferri, L., & Horcasitas, J. (2022, July 25). How to install Python 3 and set up a programming environment on an Ubuntu Server. DigitalOcean.  
<https://www.digitalocean.com/community/tutorials/how-to-install-python-3-and-set-up-a-programming-environment-on-an-ubuntu-22-04-server>
- Ubuntu 22.04 LTS – what’s new for the world’s most popular Linux desktop? | Ubuntu. (2022, April 22). Ubuntu.  
<https://ubuntu.com/blog/ubuntu-22-04-lts-whats-new-linux-desktop>