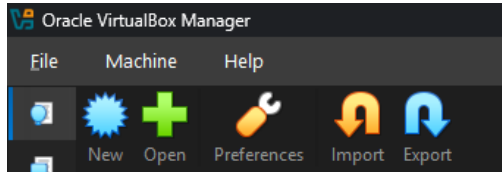


Soal 2 – Implementasi & Analisis Sistem Operasi Modern

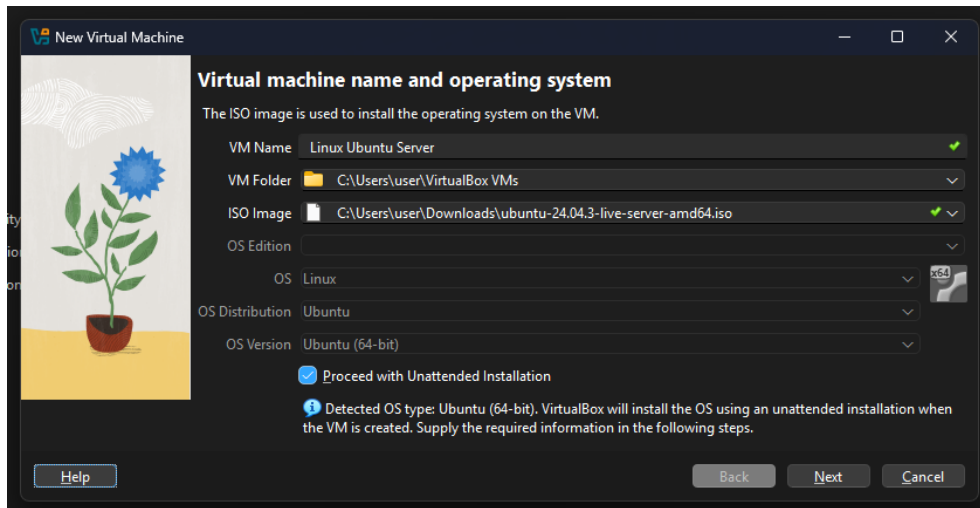
A. Virtualisasi dengan Hypervisor (VirtualBox/VMware)

POINT 1 – Install operasi system guest

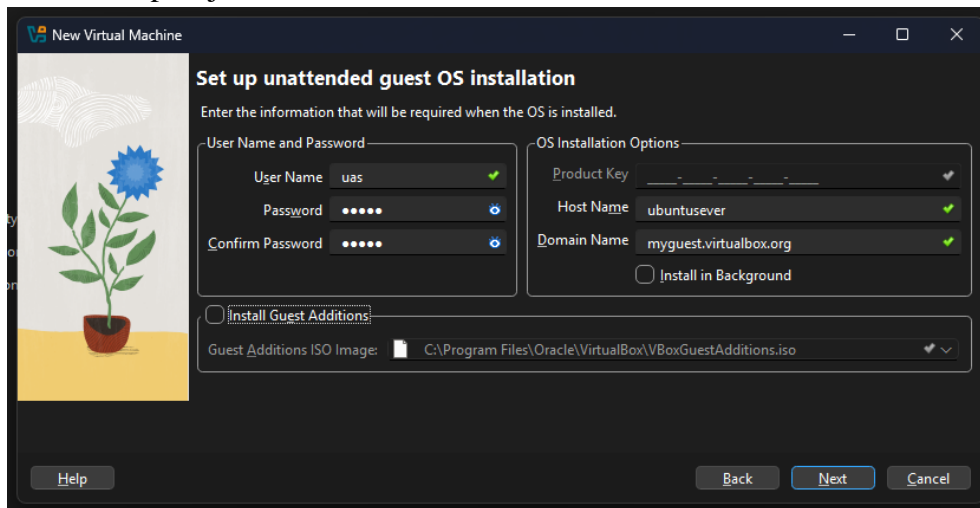
1. Pertama kalian siapkan ISO OS Linux dan Virtual Box, setelah itu kalian klik new untuk membuat virtualization baru.



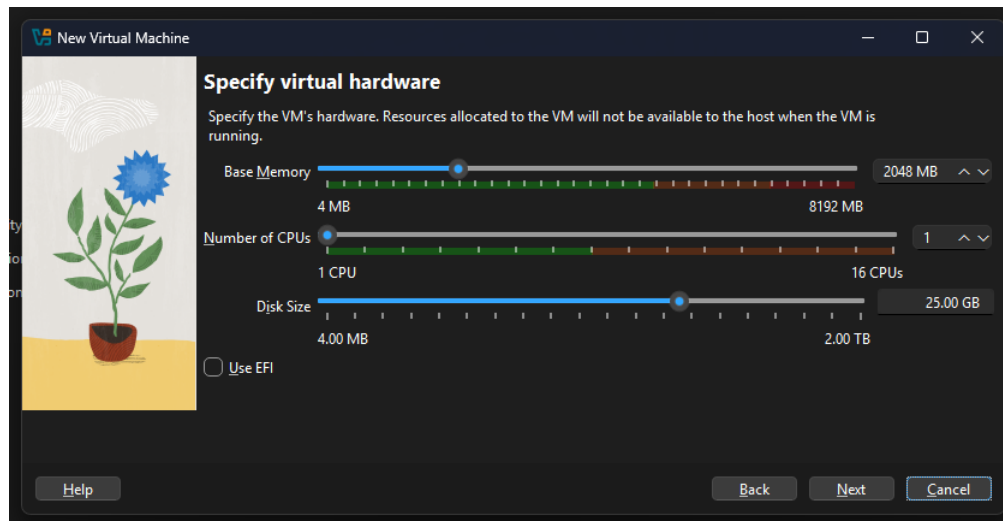
2. Kemudian kalian akan membuka pop up window dan isi nama juga file iso yang telah kalian download, kemudian klik next.



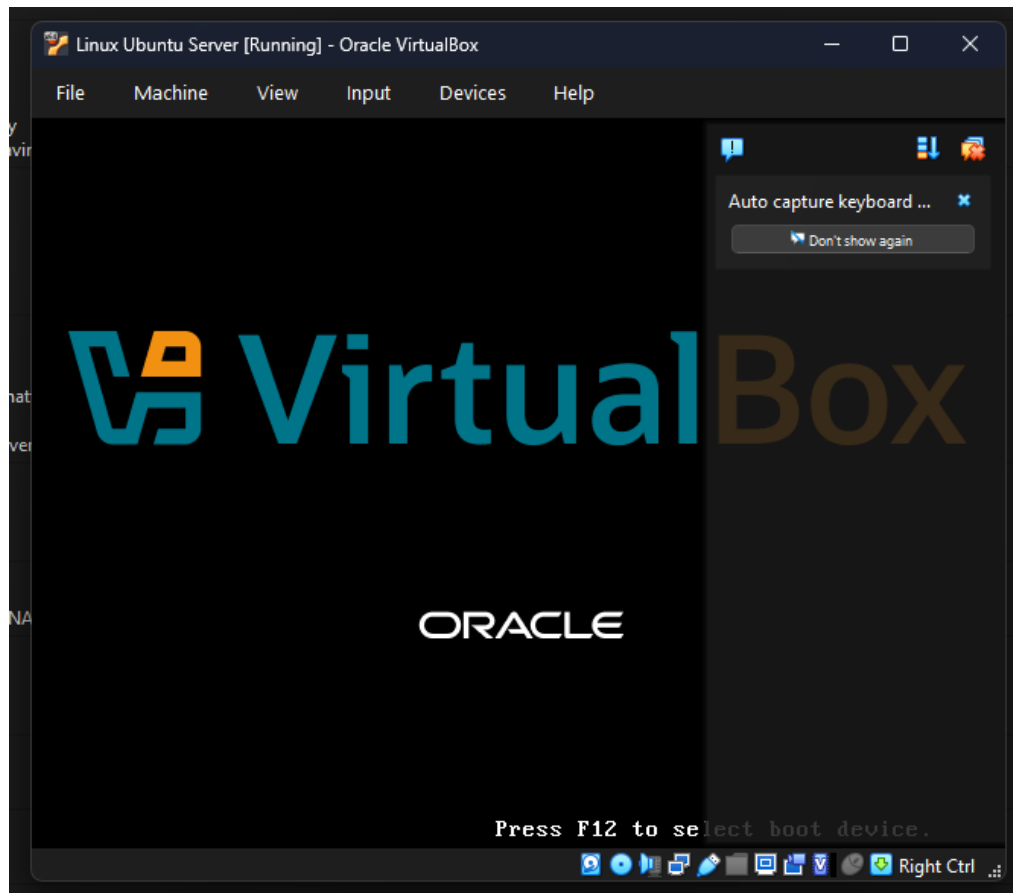
3. Kemudian kalin bisa setting setting password dan nama host, untuk nama host tidak boleh ada spasi jika sudah klik next.



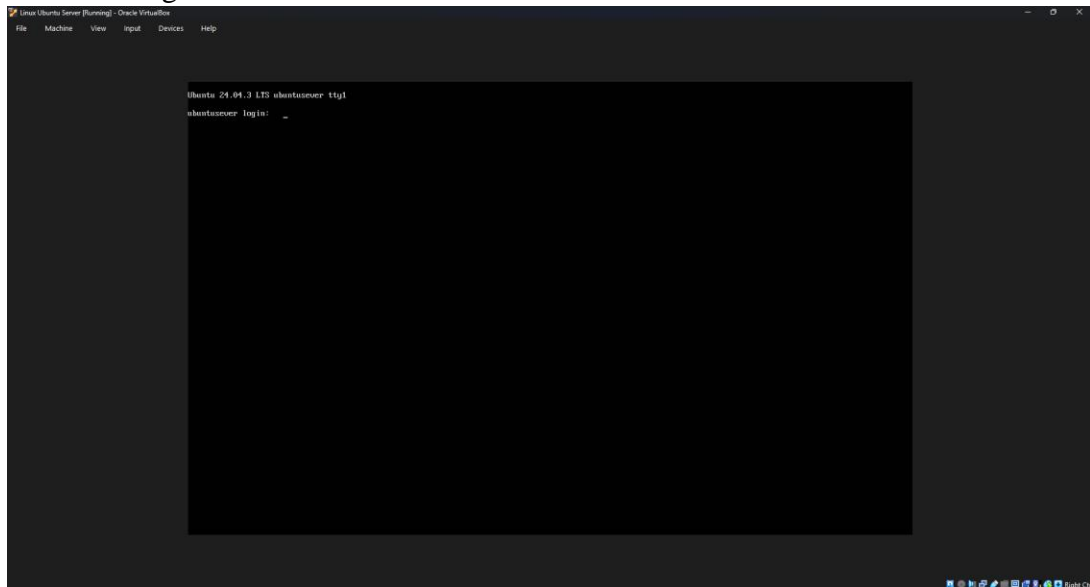
4. Atur spesifikasi virtual hardware kalian dengan kebutuhan.



5. Kemudian Ketika kalian sudah pilih next akan memunculkan pop up install linux nya.



- Setelah itu pilih yang install or try linux ubuntu dan tunggu sampai masuk ke tampilan CLI yang memerintahkan user untuk memasukkan username dan password yang sudah kalian setting di awal tadi.



- Dan ini lah tampilan pertama kali masuk linux ubuntu server.

```
Ubuntu 24.04.3 LTS ubuntu-sever tty1
ubuntu-sever login: uas
Password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-90-generic x86_64)

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

System information as of Wed Jan 28 04:56:46 AM UTC 2026

System load:          0.0
Usage of /:            10.5% of 24.44GB
Memory usage:         10%
Swap usage:            0%
Processes:             130
Users logged in:       0
IPo4 address for emp0s3: 10.0.2.15
IPo6 address for emp0s3: fd17:625c:f037:2:a00:27ff:fe16:2c0a

Expanded Security Maintenance for Applications is not enabled.

26 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

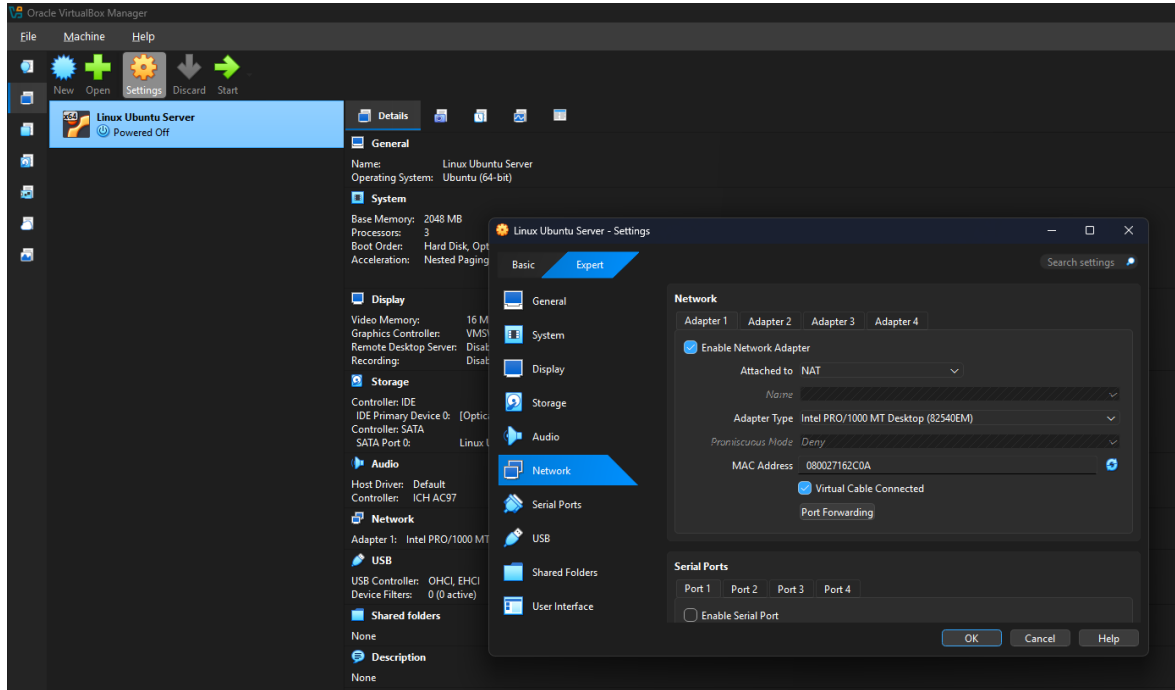
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

uas@ubuntu-sever:~$ _
```

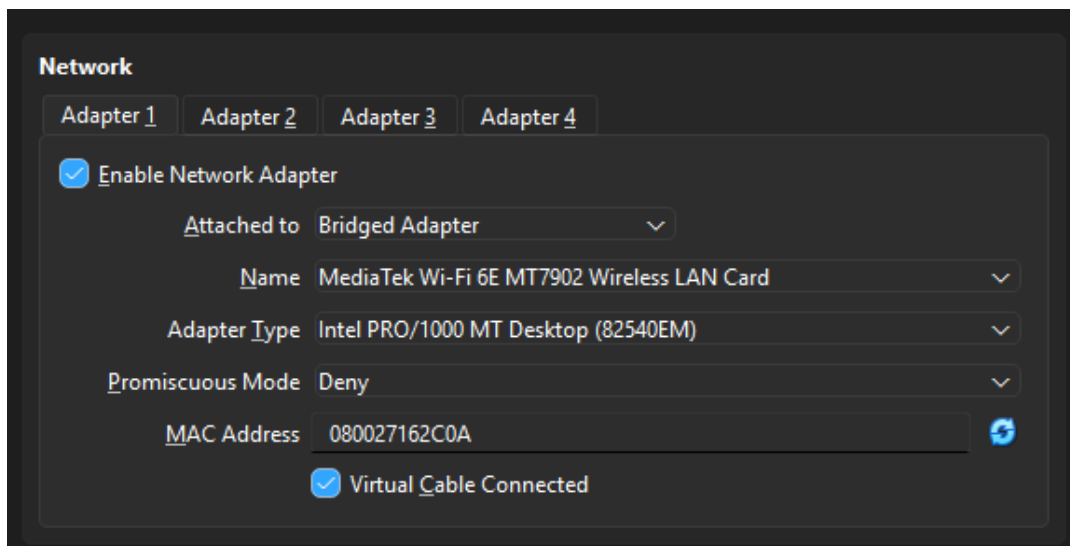
POINT 2 – Konfigurasi Jaringan (Bridged Adapter) dan sharing folder

- **Konfigurasi Jaringan**

1. Untuk konfigurasi jaringan di virtual box kalian buat masuk ke menu Setting kemudian klik Network, di kasus ini saya akan menggunakan settingan Bridged Adapter.



2. Kemudian pada bagian Attached To Ubah jadi bridged Adapter dan pilih nama wifi kalian dan yang lain default saja dan klik OK dan nyalakan VM.

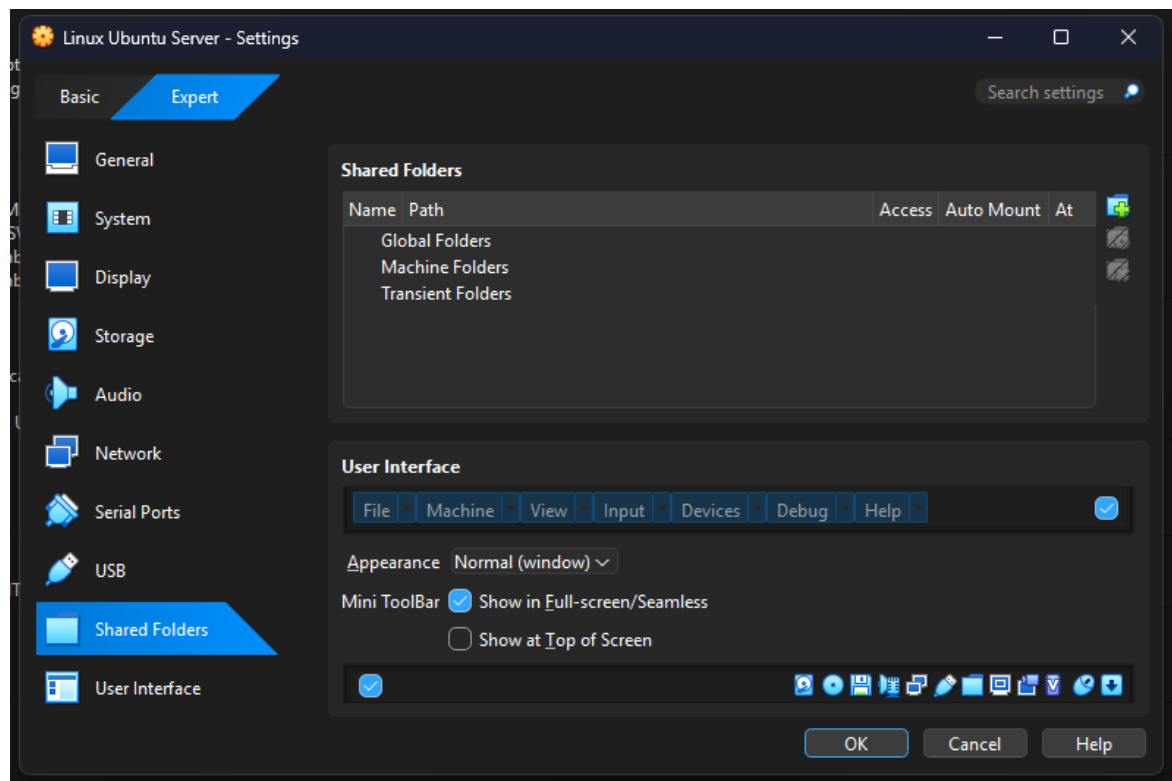


3. Dan ini adalah hasil dari setting network ke bridged adapter kita mendapatkan ip seperti ini.

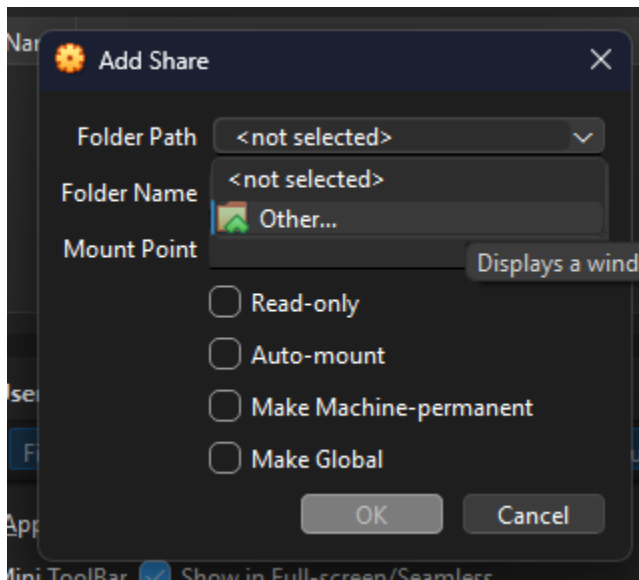
```
uas@ubuntu:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:16:2c:0a brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.6/24 metric 100 brd 192.168.1.255 scope global dynamic enp0s3
        valid_lft 86369sec preferred_lft 86369sec
    inet6 2402:8780:1003:a82:a00:27ff:fe16:2c0a/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 248314sec preferred_lft 161914sec
    inet6 fe80::a00:27ff:fe16:2c0a/64 scope link
        valid_lft forever preferred_lft forever
uas@ubuntu:~$ _
```

- **Shared Folder Guest: Virtual Box (Linux Ubuntu Server) from Host: Windows 11**

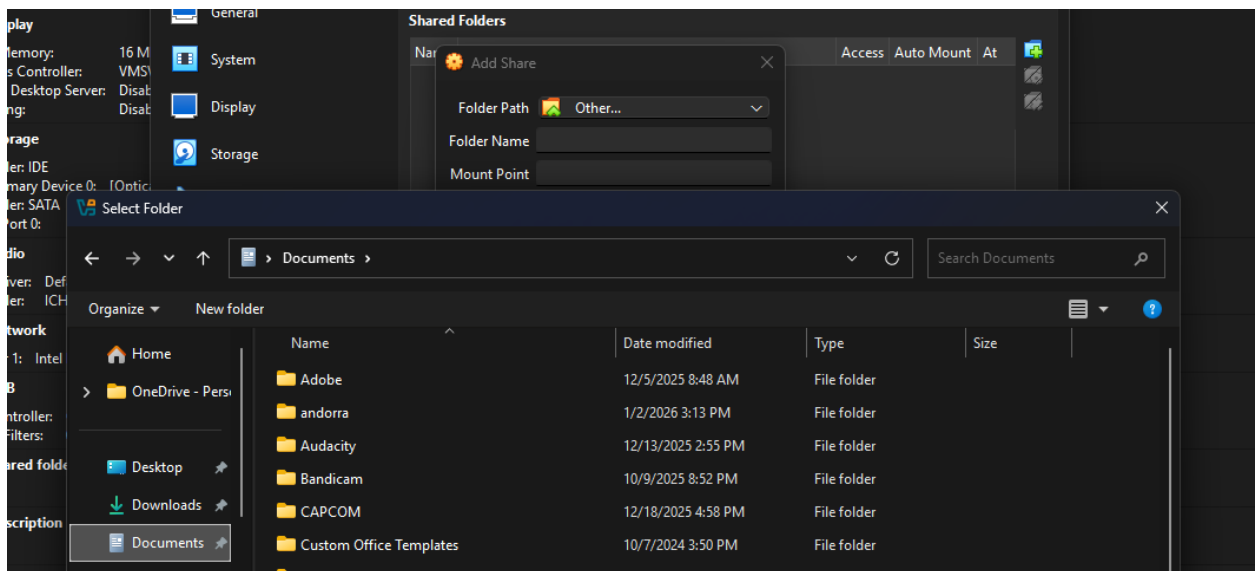
1. Untuk sharing file dari windows kalian masuk ke Setting > Shared Folder kemudian klik tanda folder biru dengan plus.



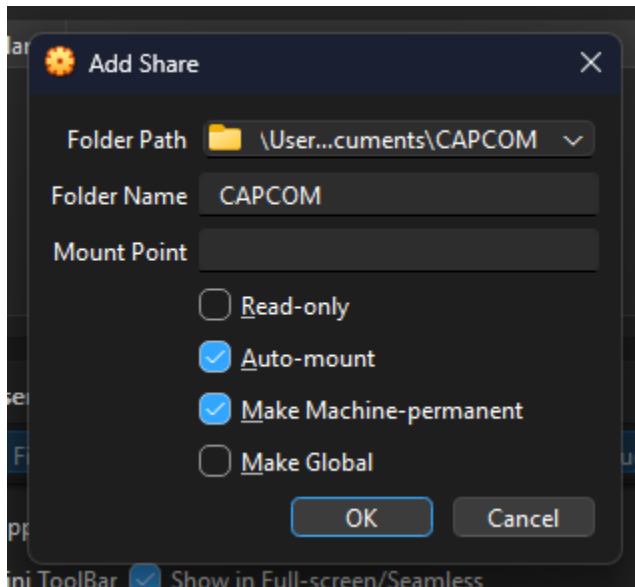
2. Kemudian ke menu folder path dan klik other.



3. Kalian pilih folder yang ingin kalian share.



- Setelah selesai centang bagian auto mount dan make machine permanent, konfigurasi ini dibuat permanen dan otomatis (Auto-mount) agar ketersediaan data antara Host dan Guest tetap terjaga meskipun sistem di-restart, kemudian klik OK.



- Kemudian kalian hidupkan VM ubuntu dan check di bagian /media karena terletak di sana lah folder shared dari windows.

```
uas@ubuntusever:~$ ls /media/
uas@ubuntusever:~$ ls -l /media/
total 0
uas@ubuntusever:~$
```

- Di case saya ini folder belum terlihat di dalam sini kemungkinan masalahnya ada di izin untuk melihat file share tersebut, dan di sini saya mencoba untuk menambahkan user saya ke vboxsf, namun pastikan dulu driver guest additions aktif dengan ketik prompt seperti ini dan tunggu install nya sampe selesai kemudian reboot.

```
uas@ubuntusever:~$ sudo apt install virtualbox-guest-utils -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  virtualbox-guest-x11
```

```
uas@ubuntusever:~$ sudo usermod -aG vboxsf $USER
uas@ubuntusever:~$ sudo reboot
```

- Silahkan check di bagian media dan kalian akan mendapati file yang kalian share sudah keluar.

```
uas@ubuntusever:~$ ls /media/
sf_CAPCOM
uas@ubuntusever:~$ _
```

POINT 3 – Hasil instalasi dan konfigurasi

1. Hasil instalasi linux ubuntu server.

```
Ubuntu 24.04.3 LTS ubuntu-sever tty1
ubuntu-sever login: uas
Password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-90-generic x86_64)

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

System information as of Wed Jan 28 04:56:46 AM UTC 2026

System load:          0.0
Usage of /:           10.5% of 24.44GB
Memory usage:         10%
Swap usage:           0%
Processes:            130
Users logged in:      0
IPo4 address for enp0s3: 10.0.2.15
IPo6 address for enp0s3: fd17:625c:f037:2:a00:27ff:fe16:2c0a

Expanded Security Maintenance for Applications is not enabled.

26 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

uas@ubuntu-sever:~$ _
```

2. Hasil konfigurasi jaringan.

```
uas@ubuntu-sever:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:16:2c:0a brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.6/24 metric 100 brd 192.168.1.255 scope global dynamic enp0s3
        valid_lft 86369sec preferred_lft 86369sec
    inet6 2402:8780:1003:a82:a00:27ff:fe16:2c0a/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 248314sec preferred_lft 161914sec
    inet6 fe80::a00:27ff:fe16:2c0a/64 scope link
        valid_lft forever preferred_lft forever
uas@ubuntu-sever:~$ _
```


B. Containerization Dengan Docker

POINT 1 – Buat dockerfile berbasis web Nginx & node.js

1. Install docker dalam linux ubuntu server yang sudah kalian install dalam virtual box tunggu sampai selesai.

```
uas@ubuntusever:~$ sudo apt install docker.io -y
```

2. Check apakah docker sudah terinstall dan berjalan dan dapat kita lihat di sini sudah aktif.

```
uas@ubuntusever:~$ sudo systemctl status docker
• docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
   Active: active (running) since Wed 2026-01-28 08:45:49 UTC; 20s ago
     TriggeredBy: • docker.socket
    Docs: https://docs.docker.com
   Main PID: 1735 (dockerd)
      Tasks: 10
     Memory: 21.6M (peak: 22.1M)
        CPU: 494ms
    CGroup: /system.slice/docker.service
            └─1735 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Jan 28 08:45:49 ubuntusever dockerd[1735]: time="2026-01-28T08:45:49.103709400Z" level=info msg="Loading containers: start."
Jan 28 08:45:49 ubuntusever dockerd[1735]: time="2026-01-28T08:45:49.510037702Z" level=info msg="Loading containers: done."
Jan 28 08:45:49 ubuntusever dockerd[1735]: time="2026-01-28T08:45:49.552164058Z" level=info msg="Docker daemon" commit="28.2.2-0ubuntu1~24.04.1" containerd-sna
Jan 28 08:45:49 ubuntusever dockerd[1735]: time="2026-01-28T08:45:49.552362556Z" level=info msg="Initializing buildkit"
Jan 28 08:45:49 ubuntusever dockerd[1735]: time="2026-01-28T08:45:49.561662281Z" level=warning msg="CDI setup error /etc/cdi: failed to monitor for changes: no
Jan 28 08:45:49 ubuntusever dockerd[1735]: time="2026-01-28T08:45:49.561715824Z" level=warning msg="CDI setup error /var/run/cdi: failed to monitor for changes
Jan 28 08:45:49 ubuntusever dockerd[1735]: time="2026-01-28T08:45:49.598617874Z" level=info msg="Completed buildkit initialization"
Jan 28 08:45:49 ubuntusever dockerd[1735]: time="2026-01-28T08:45:49.604645432Z" level=info msg="Daemon has completed initialization"
Jan 28 08:45:49 ubuntusever systemd[1]: Started docker.service - Docker Application Container Engine.
Jan 28 08:45:49 ubuntusever dockerd[1735]: time="2026-01-28T08:45:49.605543436Z" level=info msg="API listen on /run/docker.sock"

lines 1-22/22 (END)
```

3. Kemudian buat folder app masukkan command “nano index.html” dan “nano app.js” untuk isi dari index.html dan app.js itu bebas terserah kalian.

```
• uas@ubuntusever:~$ cd app/
• uas@ubuntusever:~/app$ ls
  app.js  index.html
• uas@ubuntusever:~/app$
```

4. Kemudian membuat dockerfile dengan cara memasukkan command “nano dockerfile” lalu isi dengan code yang telah di berikan di soal.

```
🔗 dockerfile
1 FROM nginx:alpine
2 COPY app /usr/share/nginx/html
3 EXPOSE 80
4 CMD ["nginx", "-g", "daemon off;"]
```

POINT 2 – Jalankan container dengan mapping port

1. Kalian bisa jalan kan perintah di bawah ini untuk Buld Docker yang telah kalian buat tadi dan tunggu sampai selesai.

```
● uas@ubuntusever:~$ sudo docker build -t uas-php .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
             Install the buildx component to build images with BuildKit:
             https://docs.docker.com/go/buildx/

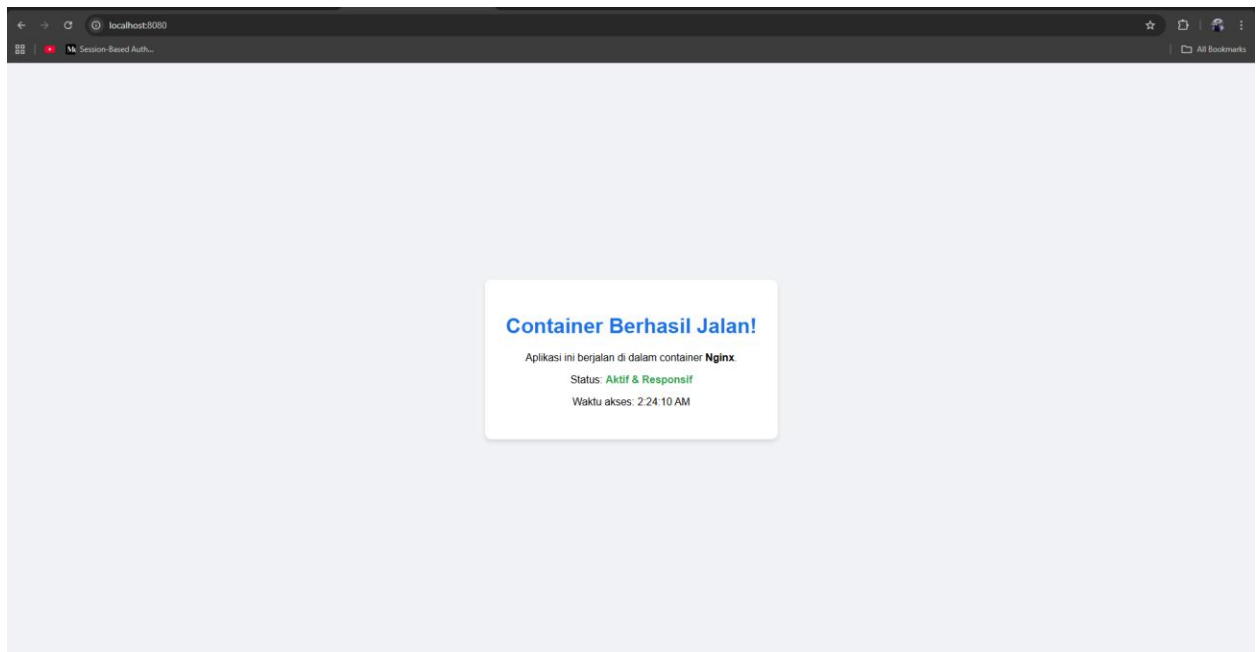
Sending build context to Docker daemon  414.2MB
Step 1/4 : FROM nginx:alpine
--> 2a855eac5070
Step 2/4 : COPY app /usr/share/nginx/html
--> Using cache
--> 41300a389157
Step 3/4 : EXPOSE 80
--> Using cache
--> c367e885c709
Step 4/4 : CMD ["nginx", "-g", "daemon off;"]
--> Using cache
--> 2c130df822fe
Successfully built 2c130df822fe
Successfully tagged uas-php:latest
```

2. Setelah itu running dengan mapping port di 8080:80 dengan perintah seperti di bawah ini dan selamat docker anda sudah jalan.

```
● uas@ubuntusever:~$ sudo docker run -d -p 8080:80 --name web-dockerfile-uas uas-php
d52ed009b336e92d53d1b4e08f8692e8b8435130678c78158474fbe0f734f90b
○ uas@ubuntusever:~$ █
```

3. kalau kurang yakin bisa di check menggunakan “sudo docker ps”.

4. Dan kalian dapat membukanya lewat host windows kalian dengan hasil seperti ini.



POINT 3 – Membuat Docker-compose.yml dengan menggabungkan 2 services web server & database (Mysql)

1. Membuat file docker-compose.yml dengan perintah “nano docker-compose.yml” dan menuliskan code ini.

```
GNU nano 7.2 docker-compose.yml *
version: '3.8'
services:
  web:
    image: uas-php
    ports:
      - "8080:80"
    depends_on:
      - db
  db:
    image: mysql:8.0
    environment:
      MYSQL_ROOT_PASSWORD: 12345
      MYSQL_DATABASE: db_tugas
```

2. Ketika kalian ingin menjalankan docker tersebut pastikan tidak ada docker yang bentrok sebelumnya. Kita bakal hapus docker run sebelum nya biar aman dengan menggunakan command seperti di bawah ini.

```
● uas@ubuntu:~$ sudo docker rm -f web-dockerfile-uas
web-dockerfile-uas
○ uas@ubuntu:~$
```

3. Kemudian running docker dengan format .yaml menggunakan command seperti di bawah ini.

```
uas@ubuntu:~$ sudo docker-compose up -d
sudo: docker-compose: command not found
uas@ubuntu:~$ _
```

4. Dan jika menemukan error seperti ini tinggal jalankan perintah “sudo apt install docker-compose -y” kemudian check menggunakan command di bawah ini.

```
uas@ubuntu:~$ docker-compose --version
docker-compose version 1.29.2, build unknown
uas@ubuntu:~$ _
```

5. Kemudian jalankan docker dengan “sudo docker-compose up -d” dan tunggu sampai selesai.

```
● uas@ubuntu:~$ sudo docker-compose up -d
Creating network "uas_default" with the default driver
Creating uas_db_1 ... done
Creating uas_web_1 ... done
```

6. Nah dan Ketika running tampilan nya bakal seperti ini.

```
● uas@ubuntu:~$ sudo docker-compose up -d
Creating network "uas_default" with the default driver
Creating uas_db_1 ... done
Creating uas_web_1 ... done
○ uas@ubuntu:~$
```

C. Analysis

Waktu Booting:

- VM : VM membutuhkan waktu booting yang cukup lama sekitar 2-4 menit, karena sistem harus inisialisasi hardware, memuat kernel, dan menjalankan berbagai layanan sistem
- Docker : untuk docker sangat cepat, hanya butuh beberapa detik karena docker tidak melakukan banyak hal yang dilakukan seperti VM, melainkan langsung menjalankan proses aplikasi,

Penggunaan Resource (RAM & CPU) :

- VM : memiliki overhead yang tinggi karena setiap VM membawa OS, RAM akan terpotong secara statis sesuai pengaturan di virtual box
- Docker : sangat efisien dan hemat resource, karena container hanya menggunakan RAM dan CPU secara dinamis sesuai dengan kebutuhan yang sedang berjalan.

Isolasi Keamanan :

- VM : memiliki keamanan yang tinggi karena dipisahkan di level hardware oleh hypervisor
- Docker : isolasi tidak sekuat VM karena semua container berbagi kernel yang sama. Celah pada salah satu container berdampak kepada seluruh Container. Meskipun docker sangat efisien, faktor keamanan tetap menjadi pertimbangan utama.