

PEMBUATAN VM DAN KONFIGURASI JARINGAN VM

Dosen Pengampu: Ferdi Chahyadi, S. Kom, M.Cs



Disusun Oleh:

Sorhan Aria Pratama	2401020008
Ikhbal Maulana	2401020039
Ferdy Afitra	2401020036
Cinto Aprilman Halawa	2401020038
Farael Ahmad	2401020023

UNIVERSITAS MARITIM RAJA ALI HAJI
FAKULTAS TEKNIK DAN TEKNOLOGI KEMARITIMAN
PROGRAM STUDI TEKNIK INFORMATIKA
2025/2026

ABSTRAK

Pada minggu ke-13, fokus kegiatan diarahkan pada proses implementasi awal hypervisor mini menggunakan QEMU/KVM, libvirt, dan virt-manager. Kegiatan mencakup verifikasi modul virtualisasi, pengujian koneksi virt-manager, pemeriksaan jaringan default libvirt, serta percobaan pembuatan virtual machine (VM). Meskipun terdapat keterbatasan lingkungan terutama karena host berjalan di VirtualBox sehingga modul KVM tidak dapat aktif proses analisis, percobaan perintah, dan validasi konfigurasi tetap dapat dilakukan. Laporan ini menyajikan tahapan implementasi, hasil pengujian, hambatan teknis, serta analisis dari sisi arsitektur dan sistem.

BAB I

PENDAHULUAN

Setelah pada minggu ke-12 dilakukan instalasi seluruh komponen hypervisor (QEMU, KVM, libvirt, dan virt-manager), minggu ke-13 merupakan tahap lanjutan berupa implementasi pembuatan VM pertama. Tujuan minggu ini adalah memastikan bahwa seluruh komponen hypervisor tidak hanya terpasang, tetapi juga berfungsi dengan benar dalam menjalankan VM.

1.1 Rumusan Masalah

1. Bagaimana membuat QEMU menggunakan QEMU/KVM dan libvirt?
2. Bagaimana memverifikasi bahwa VM dapat berjalan dan dikenali oleh libvirt?
3. Bagaimana memastikan konfigurasi jaringan virtual aktif dan dapat digunakan oleh VM?

1.2 Tujuan

1. Membuat VM menggunakan virt-manager atau perintah virt-install.
2. Mengatur parameter dasar VM (CPU, RAM, storage, jaringan).
3. Melakukan pengujian awal apakah VM dapat berjalan normal.
4. Melakukan verifikasi jaringan virtual dan koneksi VM.

BAB II

PERANCANGAN ARSITEKTUR

2.1 Arsitektur Hypervisor Mini

- a) Hypervisor Layer: QEMU + KVM + libvirt
- b) Manajemen VM: virt-manager sebagai GUI
- c) Jaringan Virtual: libvirt default network (NAT)
- d) Host: Linux yang berjalan di atas VirtualBox

2.2 Diagram Arsitektur

[Hardware] → [VirtualBox] → [Linux Host] → [QEMU/KVM] → [VM Guest]

Karena host berada dalam VirtualBox, akses langsung ke hardware virtualization tidak tersedia, sehingga KVM tidak dapat aktif.

2.3 ketebatasan arsitektur

- a) Nested virtualization tidak aktif, sehingga KVM tidak dapat memanfaatkan VT-x/AMD-V.
- b) Performa VM menurun karena QEMU berjalan tanpa akselerasi hardware.
- c) CPU usage tinggi pada host karena emulasi penuh (software virtualization).
- d) Respons VM lebih lambat dibandingkan ketika KVM aktif di bare-metal.

2.4 Desain Parameter VM

- a) CPU: 1–2 core (emulasi)
- b) RAM: 2 GB
- c) Storage: qcow2 20 GB
- d) Network: default NAT (virbr0)
- e) Boot Source: ISO installer Linux
- f) Graphics: Virtio display + Spice

2.5 Alur Implementasi Arsitektur

- a) HostLinux menjalankan libvirtd untuk manajemen hypervisor.
- b) virt-manager terhubung ke libvirt sebagai frontend GUI.
- c) Storage pool dibuat otomatis oleh libvirt.
- d) Network NAT default aktif dan digunakan oleh VM.
- e) VM dibuat menggunakan GUI atau perintah virsh/virt-install.

BAB III

IMPLEMENTASI

3.1 Pembuatan Virtual Machine

Pembuatan VM dilakukan menggunakan virt-manager dengan menentukan parameter dasar seperti sistem operasi, CPU, memori, storage, dan sumber instalasi (ISO). VM yang dibuat bertujuan sebagai VM dasar untuk pengujian lanjutan.

3.2 Verifikasi Virtual Machine, Konfigurasi dan Pemeriksaan Jaringan Virtual

Untuk memastikan VM terdaftar di libvirt, digunakan perintah:

```
virsh list --all
```

Perintah ini menampilkan daftar VM yang ada pada sistem, baik yang sedang berjalan maupun yang berhenti.:

```
virsh net-list
```

Jaringan default libvirt dengan mode NAT berada dalam status aktif, sehingga VM memiliki jalur komunikasi jaringan melalui host.

BAB IV

PENGUJIAN & ANALISIS

4.1 Pengujian Hypervisor

Pengujian yang dilakukan:

- Pemeriksaan jaringan libvirt → berhasil

```
22:54:19 ~ $ virsh list --all
Id   Name   State
-----
22:54:41 ~ $ virsh net-list
Name      State   Autostart   Persistent
-----
default   active  yes         yes
```

- Menjalankan daftar VM → berhasil

```
22:54:19 ~ $ virsh list --all
Id   Name   State
-----
22:54:41 ~ $ virsh net-list
Name      State   Autostart   Persistent
-----
default   active  yes         yes
```

- Menjalankan virt-manager → berhasil

```
virt-manager
```

- Menjalankan VM → gagal

Jika VM gagal:

- Status biasanya:
 1. shut off
 2. crashed
 3. Tidak berubah setelah diklik Start

4.2 Analisis Kegagalan

Kegagalan disebabkan oleh:

1. Host berjalan di VirtualBox, sehingga KVM tidak dapat diakses.
2. Modul `kvm_intel` atau `kvm_amd` tidak dapat dimuat karena tidak ada akses ke hardware virtualization.
3. libvirt dapat berjalan, tetapi tidak bisa memanfaatkan akselerasi hardware.

4.3 Solusi yang Disarankan

- Menggunakan Linux bare-metal (langsung tanpa VirtualBox)
- Mengaktifkan virtualization di BIOS
- Menggunakan hardware yang mendukung VT-x/AMD-V
- Menggunakan mesin lain yang mendukung KVM

BAB V

KESIMPULAN

Minggu ke-13 berfokus pada implementasi awal hypervisor mini. Meskipun VM tidak dapat dijalankan karena keterbatasan lingkungan VirtualBox yang tidak mendukung KVM, berbagai pengujian seperti koneksi virt-manager, verifikasi jaringan libvirt, serta pemeriksaan hipervisor tetap dapat terlaksana. Pemahaman mengenai mekanisme virtualisasi, struktur hypervisor, dan fungsi libvirt telah dicapai.

BAB VI

DAFTAR PUSTAKA

1. Dokumentasi resmi QEMU: <https://www.qemu.org/docs/>
2. Dokumentasi libvirt: <https://libvirt.org/>
3. Dokumentasi KVM: <https://www.linux-kvm.org/>
4. RedHat Virtualization Guide

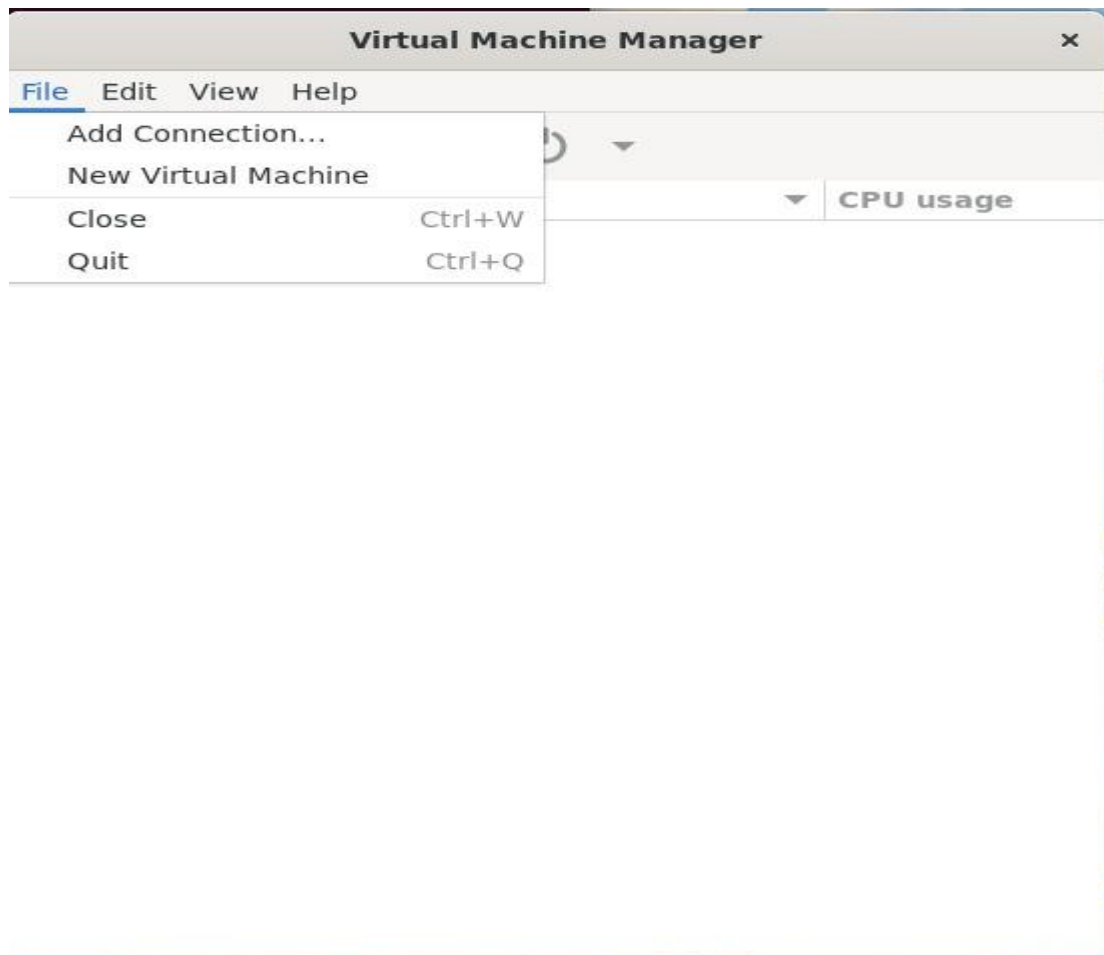
BAB VII

LAMPIRAN

5.1 Output dari perintah pembuatan VM

1. Perintah

tampilan virt-manager:



Gambar ini menunjukkan antarmuka awal virt-manager ketika baru dibuka, siap untuk membuat VM baru atau mengelola koneksi hypervisor.

2. Output tampilan lsmod:

```
sorhan@LAPTOP-372VTPNI:~$ lsmod
Module                  Size      Used by
xt_CHECKSUM             12288      1
xt_MASQUERADE           16384      3
xt_conntrack            12288      1
ipt_REJECT              12288      2
nf_reject_ipv4          12288      1 ipt_REJECT
nft_compat              16384     16
intel_rapl_msr          16384      0
intel_rapl_common       32768      1 intel_rapl_msr
kvm_intel               356352     0
crc32c_intel            16384      0
battery                 20480      0
ac                      16384      0
kvm                     970752     1 kvm_intel
irqbypass              12288      1 kvm
sch_fq_codel            16384      1
configfs                53248      0
autofs4                 45056      0
br_netfilter            28672      0
bridge                  282624     1 br_netfilter
stp                     12288      1 bridge
llc                     12288      2 bridge, stp
ip_tables               28672      0
tun                     53248      0
```

Sistem ini adalah laptop Intel dengan Linux yang sudah siap untuk virtualisasi KVM, memiliki konfigurasi jaringan yang mendukung virtualisasi/container, dan kemungkinan digunakan untuk pengembangan atau testing dengan mesin virtual.

3. Output tampilan lscpu:

```
sorhan@LAPTOP-372VTPNI:~$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Address sizes:          39 bits physical, 48 bits virtual
Byte Order:             Little Endian
CPU(s):                 16
On-line CPU(s) list:   0-15
Vendor ID:              GenuineIntel
Model name:             13th Gen Intel(R) Core(TM) i5-13450HX
CPU family:             6
Model:                 183
Thread(s) per core:     2
Core(s) per socket:     8
Socket(s):              1
Stepping:               1
BogoMIPS:               5222.40
Flags:                  fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx
                        fxsr sse2 ss ht syscall nx pdpe1gb rdtscp lm constant_tsc rep_good nopl xtopolog
                        y tsc_reliable nonstop_tsc cpuid tsc_known_freq pni pclmulqdq vmx ssse3 fma cx16 pci
                        d sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand hyp
                        ervisor lahf_lm abm 3dnowprefetch ssbd ibrs ibpb stibp ibrs_enhanced tpr_shadow ept
                        vpid ept_ad fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid rdseed adx smap clf
                        lushopt clwb sha_ni xsaveopt xsavec xgetbv1 xsaves avx_vnni vnni umip waitpkg gfni v
                        aes vpclmulqdq rdpid movdiri movdir64b fsrm md_clear serialize flush_lld arch_capabi
                        lities
```

Sistem ini memiliki CPU Intel generasi 13 yang cukup powerful (16 thread) dengan dukungan penuh virtualisasi hardware, sangat ideal untuk menjalankan KVM, container, dan workload multitasking berat.

```

Virtualization features:
  Virtualization:      VT-x
  Hypervisor vendor:   Microsoft
  Virtualization type: full
Caches (sum of all):
  L1d:                 384 KiB (8 instances)
  L1i:                 256 KiB (8 instances)
  L2:                  10 MiB (8 instances)
  L3:                  20 MiB (1 instance)
NUMA:
  NUMA node(s):        1
  NUMA node0 CPU(s):   0-15
Vulnerabilities:
  Gather data sampling: Not affected
  Itlb multihit:        Not affected
  L1tf:                 Not affected
  Mds:                  Not affected
  Meltdown:             Not affected
  Mmio stale data:      Not affected
  Reg file data sampling: Mitigation; Clear Register File
  Retbleed:             Mitigation; Enhanced IBRS
  Spec rstack overflow: Not affected
  Spec store bypass:    Mitigation; Speculative Store Bypass disabled via prctl
  Spectre v1:           Mitigation; usercopy/swapgs barriers and __user pointer sanitization
  Spectre v2:           Mitigation; Enhanced / Automatic IBRS; IBPB conditional; RSB filling; PBRSB-eIBRS SW
                        sequence; BHI BHI_DIS_S
  Srbds:                Not affected
  Tsx async abort:      Not affected

```

Sistem ini memiliki CPU Intel modern dengan performa tinggi, dilengkapi dukungan virtualisasi hardware lengkap dan fitur keamanan mutakhir. Sangat cocok untuk:

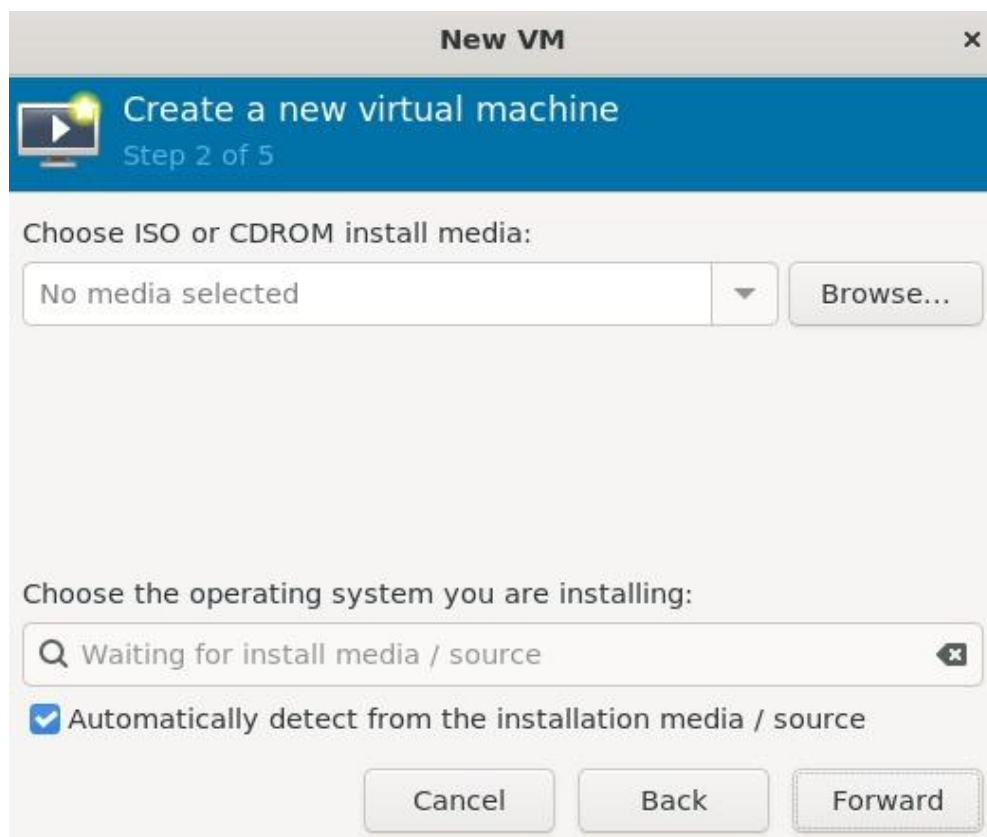
- Menjalankan mesin virtual (KVM/QEMU) dengan performa near-native
- Development dan testing lingkungan virtual/container
- Workload komputasi berat seperti data processing, machine learning, atau server development lokal.

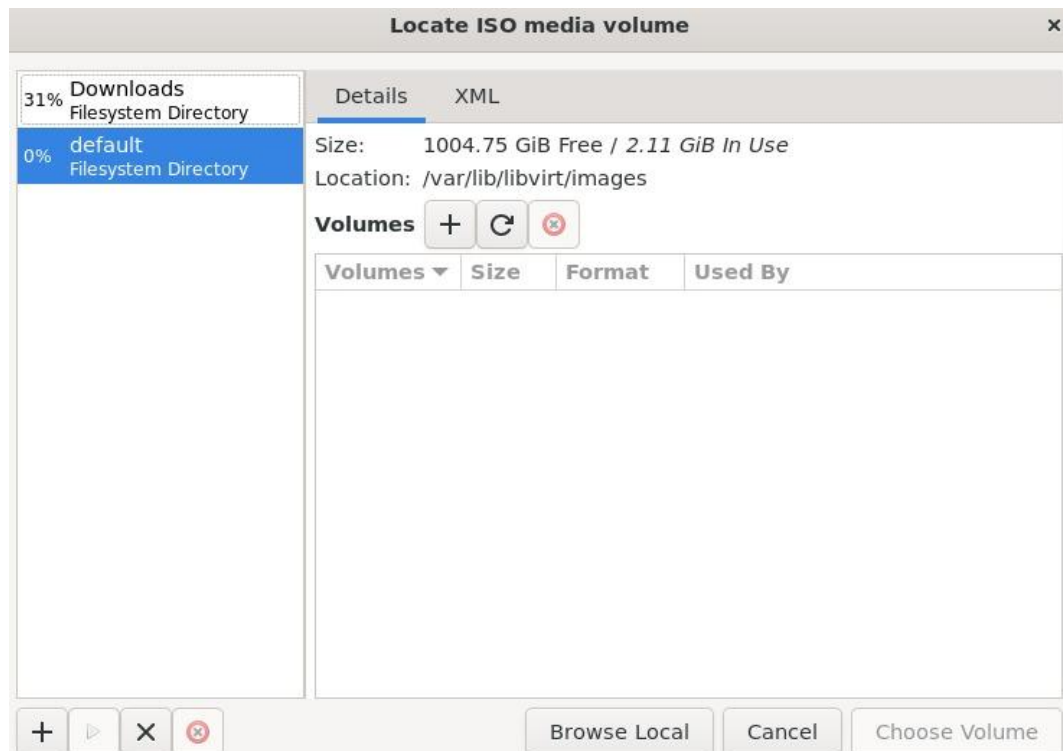
4. Tampilan dari pembuatan VM
Step 1:



Gambar ini menunjukkan tampilan awal pembuatan VM di virt-manager, siap untuk memilih metode instalasi OS sebelum melanjutkan ke konfigurasi resource (CPU, RAM, storage, dll.).

Step 2:





Recent	Name	Location	Size	Type	Accessed
Home	ubuntu-24.04.3-desktop-amd64.iso	/mnt/d/Dataku/Downloads	6.3 GB	raw CD image	11:04
Other Locations					

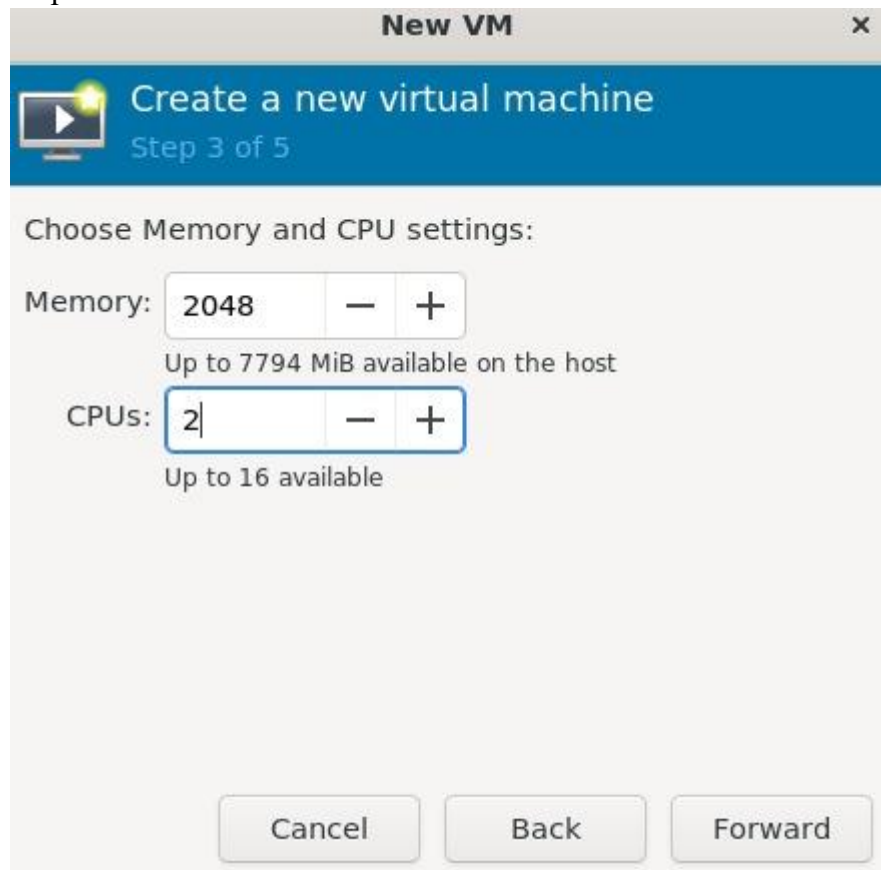
Penjelasan:

- Langkah 2 dari 5: Pengguna sedang memilih media instalasi OS (ISO image) untuk mesin virtual.
- Hypervisor: Masih menggunakan QEMU/KVM dengan penyimpanan default di /var/lib/libvirt/images.
- Pemilihan ISO:
 - Pengguna telah menelusuri direktori dan menemukan file ubuntu-24.04.3-desktop-amd64.iso (Ukuran 6.3 GB) di lokasi /mnt/d/Dataku/Downloads.
 - SO ini adalah Ubuntu 24.04.3 Desktop versi 64-bit.
- Akan Dilakukan: Setelah memilih ISO tersebut, pengguna akan melanjutkan ke langkah konfigurasi berikutnya (seperti alokasi RAM, CPU, dan storage) untuk menginstal Ubuntu sebagai sistem operasi guest di mesin virtual.

Konteks:

Ini adalah proses standar pembuatan VM di virt-manager, di mana pengguna memilih ISO lokal untuk instalasi OS guest (dalam hal ini Ubuntu 24.04), sebelum mengonfigurasi resource hardware virtual.

Step 3:



New VM

Create a new virtual machine
Step 3 of 5

Choose Memory and CPU settings:

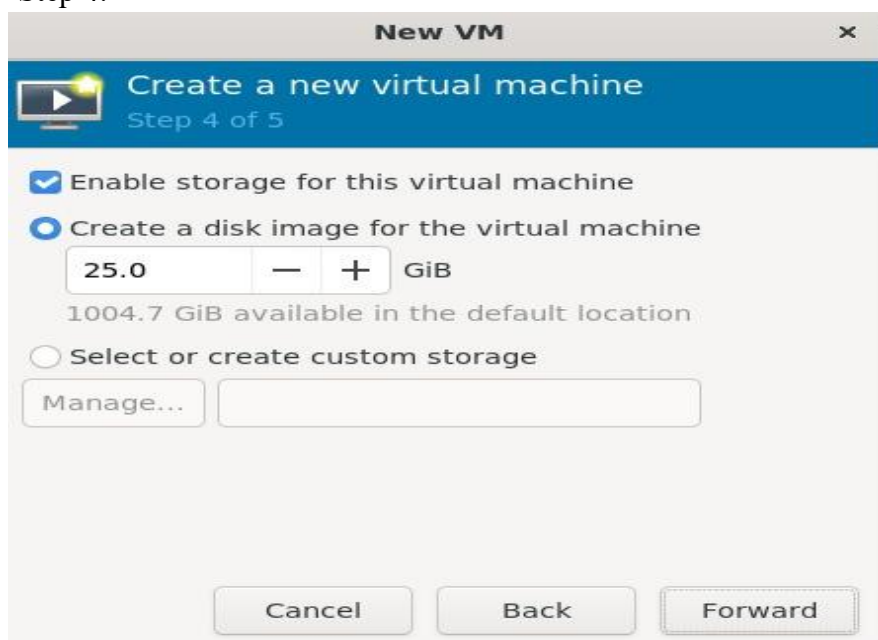
Memory: 2048 — +
Up to 7794 MiB available on the host

CPUs: 2 — +
Up to 16 available

Cancel Back Forward

Pengguna sedang mengalokasikan resource RAM dan CPU untuk mesin virtual Ubuntu yang akan dibuat. Dengan host yang cukup kuat (16 vCPU, ~7,8 GB RAM tersedia), pengguna memiliki fleksibilitas untuk memberi resource lebih jika diperlukan. Langkah selanjutnya biasanya pengaturan storage (disk virtual).

Step 4:



New VM

Create a new virtual machine
Step 4 of 5

☒ Enable storage for this virtual machine

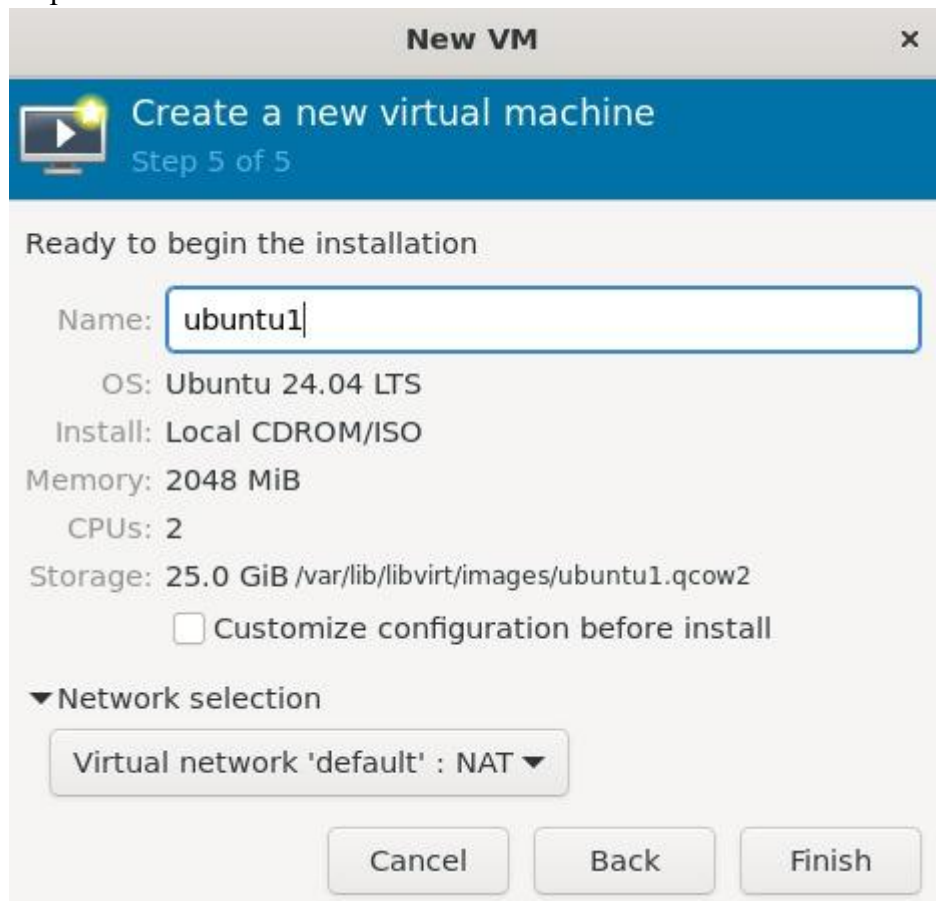
☒ Create a disk image for the virtual machine
25.0 — + GiB
1004.7 GiB available in the default location

☐ Select or create custom storage
Manage...

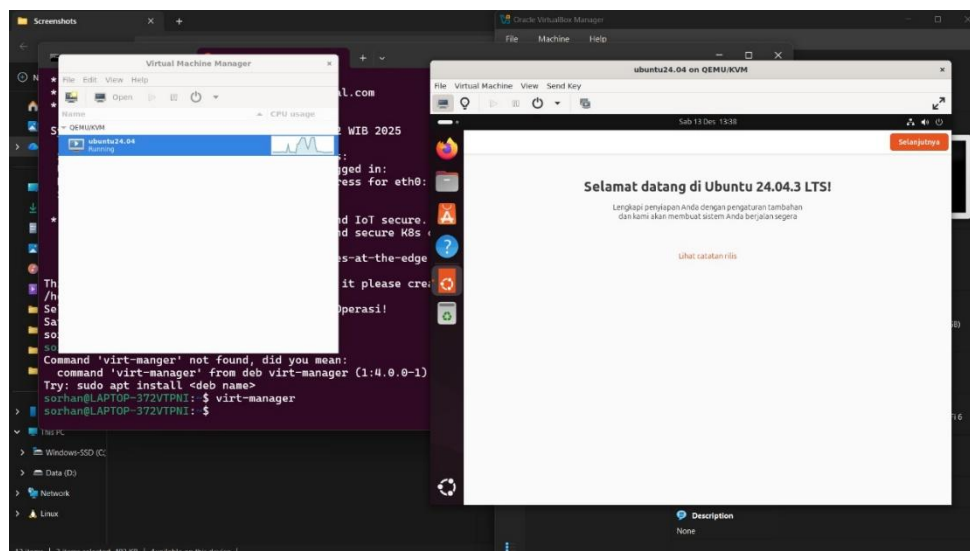
Cancel Back Forward

Pengguna sedang mengonfigurasi disk virtual untuk mesin virtual Ubuntu. Disk baru sebesar 25 GB akan dibuat di lokasi default. Langkah selanjutnya (step 5) biasanya adalah penamaan VM dan review konfigurasi sebelum membuat VM.

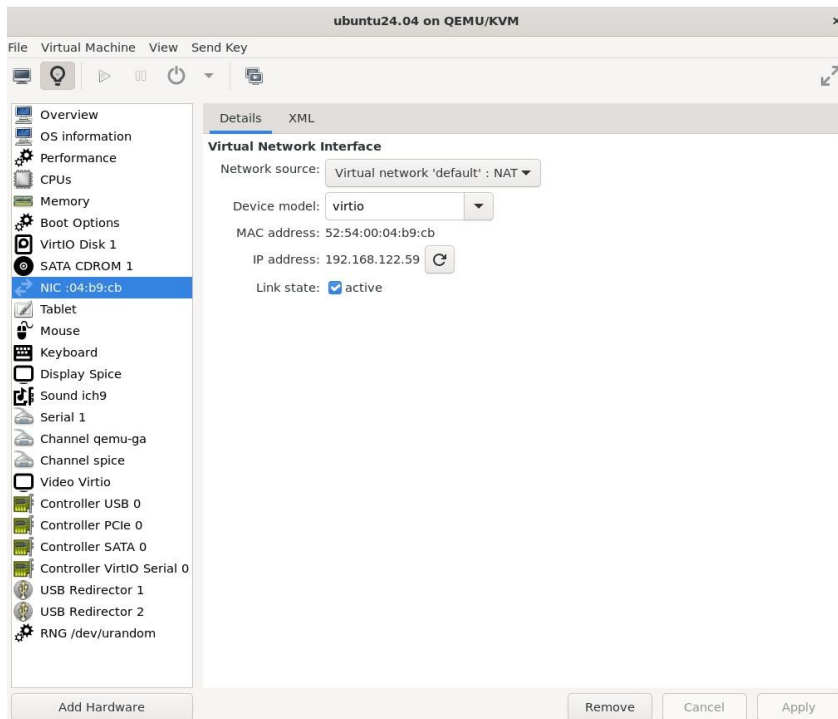
Step 5:



Menampilkan beberapa jendela aplikasi yang terbuka di desktop yang utamanya berkaitan dengan virtualisasi:



Konfigurasi Jaringan VM:



```
ggs@ggs-Standard-PC-Q35-ICH9-2009: ~  
valid_lft forever preferred_lft forever  
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr  
oup default qlen 1000  
    link/ether 52:54:00:04:b9:cb brd ff:ff:ff:ff:ff:ff  
    inet 192.168.122.59/24 brd 192.168.122.255 scope global dynamic noprefixrou  
t e enp1s0  
        valid_lft 3502sec preferred_lft 3502sec  
        inet6 fe80::5054:ff:fe04:b9cb/64 scope link  
        valid_lft forever preferred_lft forever  
ggs@ggs-Standard-PC-Q35-ICH9-2009:~$ ping -c 4 google.com  
PING google.com (142.251.10.113) 56(84) bytes of data:  
64 bytes from sd-in-f113.1e100.net (142.251.10.113): icmp_seq=1 ttl=104 time=325  
ms  
64 bytes from sd-in-f113.1e100.net (142.251.10.113): icmp_seq=2 ttl=104 time=108  
1 ms  
64 bytes from sd-in-f113.1e100.net (142.251.10.113): icmp_seq=3 ttl=104 time=228  
ms  
64 bytes from sd-in-f113.1e100.net (142.251.10.113): icmp_seq=4 ttl=104 time=285  
ms  
--- google.com ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3000ms  
rtt min/avg/max/mdev = 227.786/479.805/1081.183/348.915 ms, pipe 2  
ggs@ggs-Standard-PC-Q35-ICH9-2009:~$
```

Ini adalah tampilan setelah instalasi selesai, di mana VM sudah berjalan dengan sistem operasi Ubuntu 24.04 yang berjalan normal, terhubung ke internet, dan siap digunakan. Pengguna dapat melihat detail hardware virtual dan melakukan manajemen lebih lanjut melalui antarmuka virt-manager.