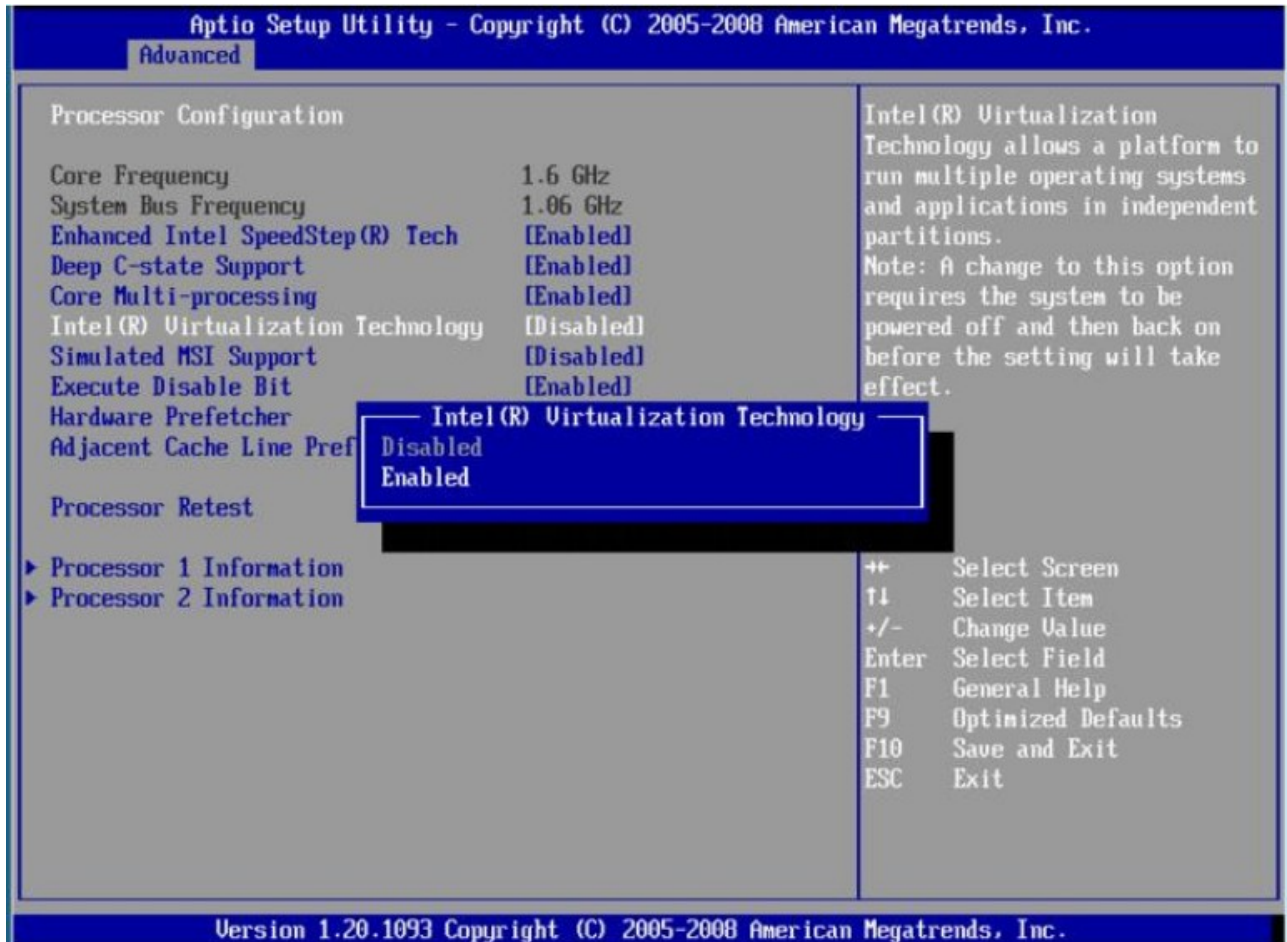


## PRAKTIKUM VIRTUALISASI

### Lab 1 Mengaktifkan dukungan virtualisasi hardware pada CPU

- Agar virtualisasi KVM bekerja, hardware harus mendukung virtualisasi. KVM membutuhkan komitmen penuh dari CPU untuk melakukan virtualisasi. Meskipun hardware Anda mendukung virtualisasi KVM, Anda harus mengaktifkan fitur ini di BIOS komputer Anda untuk memanfaatkan kemampuan ini. Silahkan lakukan pemeriksaan fitur teknologi virtualisasi pada BIOS Anda, dengan melakukan SETUP BIOS kemudian aktifkan fitur tersebut. (Lihat gambar 1).



Gambar 1: Mengaktifkan Fitur Virtualisasi di BIOS

### Lab 2 Memeriksa dukungan virtualisasi pada CPU

- Untuk membuktikan jika dukungan teknologi virtualisasi pada cpu telah diaktifkan, lakukan pemeriksaan dengan perintah berikut di terminal Linux Anda:  

```
$ egrep -c '(vmx|svm)' /proc/cpuinfo
```

  
4  
Jika outputnya menghasilkan nilai lebih dari 0 maka cpu sudah mendukung teknologi virtualisasi.

### Lab 3 Memeriksa dukungan CPU 64 bit

- Untuk membuktikan jika cpu telah mendukung pemrosesan 64 bit, lakukan pemeriksaan dengan perintah berikut di terminal Linux Anda:  

```
$ lscpu | grep op-mode
```

  
Jika cpu pada komputer Anda mendukung pemrosesan 64 bit, maka akan tampak output seperti berikut:  
CPU op-mode(s): 32-bit, 64-bit

- Anda dapat juga memeriksa dukungan cpu 64 bit melalui proc filesystem dengan perintah berikut:  
`$ egrep -c 'lm' /proc/cpuinfo`  
4  
Jika outputnya menghasilkan nilai lebih dari 0 maka cpu sudah mendukung pemrosesan 64 bit.

#### Lab 4 Memeriksa dukungan Linux OS 64 bit

- KVM membutuhkan host os sebagai host yang akan menjalankan hypervisor KVM menggunakan kernel Linux 64 bit. Untuk itu periksalah apakah kernel linux yang digunakan pada komputer Anda saat ini adalah kernel Linux untuk cpu 64 bit, dengan perintah berikut ini:  
`$ uname -m`  
atau  
`$ uname -p`  
atau  
`$ lscpu | grep -i arch`
- Jika kernel Linux yang Anda gunakan adalah kernel 64 bit maka perintah-perintah di atas akan menghasilkan output sebagai berikut:  
`$ uname -m`  
x86\_64  
`$ uname -p`  
x86\_64  
`$ lscpu | grep -i arch`  
Architecture: x86\_64

#### Lab 5 Melakukan instalasi virtualisasi KVM

- Untuk melakukan instalasi perangkat lunak virtualisasi KVM, lakukan perintah berikut ini:  
`# apt-get install qemu-kvm libvirt-bin virtinst`
- Selanjutnya pastikan service libvirtd sudah berjalan agar KVM dapat digunakan, lakukan perintah berikut ini:  
`# systemctl status libvirtd`
- Apabila service libvirtd belum berjalan, lakukan perintah berikut ini untuk menjalankannya:  
`# systemctl start libvirtd`

#### Lab 6 Mengelola Virtual Machine dengan command line

##### Lab 6.1 Membuat Virtual Machine

- Untuk membuat virtual machine menggunakan command line, ketikkan perintah berikut:  
`# virt-install \`  
`--name Ubuntu-18.04 \`  
`--memory=1024 \`  
`--vcpus=1 \`  
`--disk size=20 \`  
`--cdrom /home/rangga/Downloads/ubuntu-18.04.1-server-amd64.iso \`  
`--virt-type=qemu \`  
`--network bridge=virbr0 \`  
`--graphics vnc,password=rahasia,port=5900,listen=0.0.0.0`

Keterangan:

- `--name=nama_vm` → nama virtual machine
- `--memory=besar_ram` → jumlah memory yang digunakan untuk virtual machine
- `--vcpus=jumlah`, jumlah virtual CPU yang digunakan
- `--disk size=besar` → besar disk dalam GB
- `--cdrom /path/to/file.iso` → device file image CD

- `--virt-type=qemu` → tipe dari virtual machine QEMU
- `--network bridge=virbr0` → tipe dari jaringan yang digunakan adalah bridge
- `--graphics vnc,password=rahasia,port=5900,listen=0.0.0.0` → tampilan menggunakan protokol VNC dengan password rahasia, berjalan pada port 5900 dan listening pada semua interface

### Lab 6.2 Menampilkan daftar Virtual Machine

- Untuk menampilkan daftar virtual machine, gunakan perintah berikut:  
# `virsh list --all`

### Lab 6.3 Menjalankan dan Mengakses Virtual Machine

- Untuk menjalankan virtual machine yang sudah dibuat menggunakan command line, ketikkan perintah berikut:  
# `virsh start Ubuntu-18.04`
- Untuk mengakses virtual machine yang sudah dibuat menggunakan command line, ketikkan perintah berikut:  
# `virt-viewer Ubuntu-18.04`

### Lab 6.4 Mengakses Virtual Machine dari komputer lain

- Untuk mengakses virtual machine yang sudah dibuat dari komputer lain dapat menggunakan aplikasi remmina. Berikut langkah-langkahnya:
  1. Buka aplikasi Remmina dengan menjalankan “remmina” pada terminal, atau memilihnya pada daftar aplikasi di Ubuntu
  2. Klik tombol New
  3. Masukkan sbb:
    - a. Name, diisikan nama koneksi
    - b. Protocol, diisi dengan VNC
    - c. Server, diisi dengan alamat dari host parent dari virtual machine, bukan virtual machine-nya sendiri. Kemudian dilanjutkan dengan nomor port dari VNC. Contoh penulisan:  
`<ip_parent>:<port_vnc>`
    - d. Color depth, kedalaman warna dari layar. Bila tampilan grafis tidak terlalu penting sebaiknya dipilih 256 colors untuk menghemat bandwidth.
    - e. Poor, kualitas gambar. Bila tampilan grafis tidak terlalu penting sebaiknya dipilih Poor untuk menghemat bandwidth.
  4. Klik tombol Connect
  5. Maka akan diminta password, maka masukkan password sesuai dengan password yang diberikan saat instalasi.

### Lab 6.5 Mengubah Virtual Machine

#### Lab 6.5.1 Mematikan virtual machine

- Sebelum virtual machine dapat diubah maka virtual machine yang sedang berjalan harus dimatikan terlebih dahulu. Gunakan perintah berikut:  
# `virsh destroy Ubuntu-18.04`

#### Lab 6.5.2 Mengubah RAM dan Virtual CPU

- Untuk mengubah RAM dan virtual CPU, jalankan perintah berikut:  
# `virsh edit Ubuntu-18.04`  

```
<!-- ubah besar RAM pada memory unit dan currentMemory unit -->
<memory unit='KiB'>2097152</memory>
<currentMemory unit='KiB'>2097152</currentMemory>
<!-- ubah besar jumlah virtual CPU pada vcpu -->
<vcpu placement='static'>2</vcpu>
```

### Lab 6.5.3 Menambah dan Melepas Media Penyimpanan

- Untuk membuat image harddisk sebagai harddisk baru dalam virtual machine dengan besar 5 GB, gunakan perintah berikut:  
`# qemu-img create -f qcow2 hd2.qcow2 5G`
- Untuk memasang media penyimpanan saat virtual machine sedang berjalan Anda dapat menggunakan perintah berikut:  
`# virsh attach-disk <nama_vm> --source /path/dari/image atau device --target <nama_device>`

Contoh menghubungkan device flashdisk ke virtual machine

```
# virsh attach-disk Ubuntu-18.04 --source /dev/sdb1 --target vdb
```

Contoh menghubungkan device image harddisk secara permanen

```
# virsh attach-disk Ubuntu-18.04 --source /old/virtual/hd2.img --target vdc --persistent
```

- Untuk melepas harddisk bila tidak diperlukan  
`# virsh detach-disk <nama_vm> --target <nama_device>`

Contoh:

```
# virsh detach-disk Ubuntu-18.04 --target vdb
```

### Lab 6.6 Mengatur Virtual Machine Berjalan saat Boot

- Untuk mengkonfigurasi agar suatu virtual machine secara otomatis berjalan ketika komputer host booting, gunakan perintah berikut:  
`# virsh autostart Ubuntu-18.04`
- Untuk menonaktifkan virtual machine berjalan secara otomatis pada saat booting, gunakan perintah berikut:  
`# virsh autostart Ubuntu-18.04 --disable`

### Lab 6.7 Mematikan dan menghapus Virtual Machine

- Untuk mematikan virtual machine yang sedang berjalan, gunakan perintah berikut:  
`# virsh destroy Ubuntu-18.0`
- Untuk menghapus virtual machine, gunakan perintah berikut:  
`# virsh undefine Ubuntu.18.04`