

ONE DAY WORKSHOP



**PROXMOX
AUTOMATION
WITH
ANSIBLE**

I PUTU HARIYADI

WWW.UNIVERSITASBUMIGORA.AC.ID

**MODUL ONE DAY WORKSHOP
PROXMOX AUTOMATION WITH ANSIBLE**



**OLEH
I PUTU HARIYADI**

**UNIVERSITAS BUMIGORA
www.universitasbumigora.ac.id**

KATA PENGANTAR

Puji syukur penyusun panjatkan kepada Tuhan Yang Maha Esa atas berkat dan rahmatnya sehingga “**MODUL ONE DAY WORKSHOP PROXMOX AUTOMATION WITH ANSIBLE**” ini dapat terselesaikan. Modul ini dibuat sebagai panduan bagi peserta workshop yang diadakan oleh **Unit Kegiatan Mahasiswa (UKM) Network Community (NETCOM) Universitas Bumigora**, pada hari Sabtu dan Minggu, 2-3 Nopember 2019.

Penyusun menyadari bahwa modul workshop ini masih jauh dari sempurna. Untuk itu kritik dan saran demi pengembangan modul workshop ini sangat diharapkan. Kritik dan saran dapat dikirimkan melalui email dengan alamat: *putu.hariyadi@universitasbumigora.ac.id*.
Terimakasih.

Mataram, 30 Oktober 2019

Penyusun

DAFTAR ISI

HALAMAN JUDUL	(i)
KATA PENGANTAR	(ii)
DAFTAR ISI	(iii)
PENDAHULUAN	(1)
BAB I INSTALASI DAN KONFIGURASI PROXMOX VE 5.3 PADA VMWARE WORKSTATION 14	(2)
BAB II MENONAKTIFKAN PESAN NOTIFIKASI “NO VALID SUBSCRIPTION” PADA PROXMOX VE 5.3	(26)
BAB III MENONAKTIFKAN PVE ENTERPRISE SUBSCRIPTION DAN MENGAKTIFKAN PVE NO-SUBSCRIPTION REPOSITORY PADA PROXMOX VE 5.3	(31)
BAB IV INSTALASI DAN KONFIGURASI MIKROTIK CHR SEBAGAI INTERNET GATEWAY PADA VMWARE WORKSTATION 15	(35)
BAB V INSTALASI DAN KONFIGURASI LINUX CONTAINER (LXC) CENTOS 7 PADA PROXMOX VE 5.3	(52)
BAB VI MANAJEMEN USER DAN PERMISSION PADA PROXMOX VE 5.3	(64)
BAB VII PROXMOX VE 5.3 APPLICATION PROGRAMMING INTERFACE (API)	(77)
BAB VIII PENGENALAN ANSIBLE DAN ANSIBLE PROXMOX MODULE	(95)
BAB IX INSTALASI DAN KONFIGURASI LINUX CENTOS 7 SEBAGAI ANSIBLE CONTROL MACHINE PADA VMWARE WORKSTATION 14	(102)
BAB X ANSIBLE INVENTORY, AD-HOC DAN PLAYBOOK SERTA	

ANSIBLE PROXMOX_TEMPLATE MODULE	(140)
BAB XI STUDI KASUS OTOMASI MANAJEMEN CONTAINER, USER DAN PERMISSION DARI PVE DENGAN ANSIBLE PLAYBOOK	(161)
DAFTAR REFERENSI	(185)
TENTANG PENULIS	(186)

PENDAHULUAN

Adapun kebutuhan perangkat keras (*hardware*) dan lunak (*software*) yang diperlukan untuk dapat mengujicoba materi yang terdapat pada modul *workshop* ini adalah sebagai berikut:

A. Kebutuhan *Hardware*

Satu unit komputer dengan rekomendasi spesifikasi sebagai berikut:

1. CPU: 64 bit.
2. RAM: 8 GB.
3. Hard drive.
4. 1 (satu) *Network Interface Card*.

B. Kebutuhan *Software*

1. *Proxmox Virtual Environment (VE) versi 5.3* yang dapat diunduh pada situs *Proxmox* di alamat <https://www.proxmox.com/en/downloads>
2. *VMWare Workstation 14 Pro* atau *15 Pro*.
3. *Putty SSH Client* yang dapat diunduh pada alamat <https://www.putty.org/>
4. *Browser Chrome* yang dapat diunduh pada alamat <https://www.google.com/chrome/>.
5. *Mikrotik Cloud Hosted Router (CHR)* dan *Winbox* yang dapat diunduh pada situs *Mikrotik* pada alamat <https://mikrotik.com/download>
6. *Linux Container Image Templates* untuk *Proxmox* yang dapat diunduh pada alamat <http://download.proxmox.com/images/system/>
7. *CentOS 7* sebagai *Ansible Automation Server* bagi *Proxmox VE 5.3* dan dapat diunduh pada alamat <https://www.centos.org/>.

Selain itu juga diperlukan koneksi *Internet* untuk mengunduh perangkat lunak tersebut dan ujicoba materi.

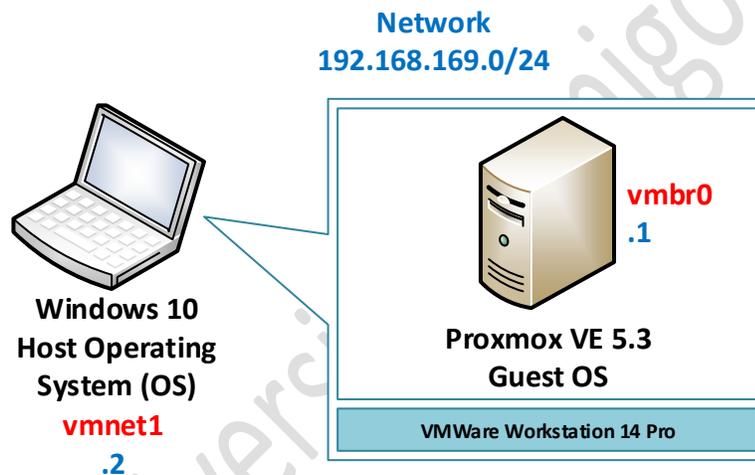
BAB I

INSTALASI DAN KONFIGURASI PROXMOX VE 5.3

PADA VMWARE WORKSTATION 14

A. Rancangan Jaringan Ujicoba

Rancangan jaringan ujicoba terdiri dari 1 unit *notebook* dengan sistem operasi *Windows 10* yang telah diinstalasi *VMWare Workstation Pro 14* sebagai *hosted hypervisor*, seperti terlihat pada gambar berikut:



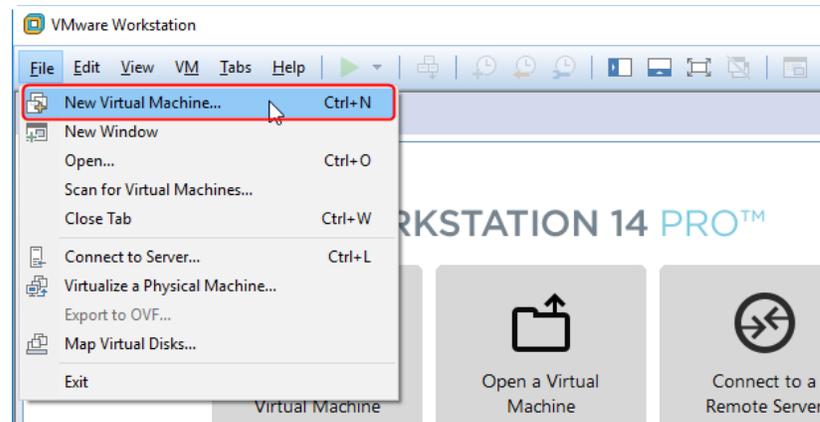
Pada *VMWare Workstation* akan dibuat *Guest Virtual Machine* dengan *Operating System (OS)* *Proxmox VE 5.3*. Alamat jaringan yang digunakan adalah $192.168.169.0/24$ dengan alokasi pengalamatan IP meliputi $192.168.169.1$ untuk interface **vmbr0** di *Guest OS Proxmox* dan $192.168.169.2$ untuk interface **vmnet1** di *Windows 10*.

B. Instalasi Proxmox VE 5.3

Adapun langkah-langkah instalasi dan konfigurasi *Proxmox VE 5.3* pada *VMWare Workstation 14 Pro* adalah sebagai berikut:

1. Jalankan aplikasi *VMWare Workstation 14 Pro* melalui **Start > VMWare > VMWare Workstation Pro**.

2. Tampil aplikasi *VMWare Workstation*. Untuk membuat *virtual machine* baru, pilih menu **File > New Virtual Machine ...**, seperti terlihat pada gambar berikut:



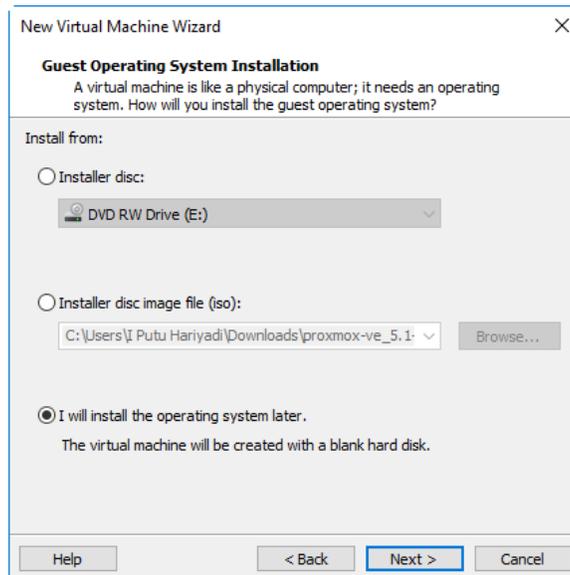
3. Tampil kotak dialog *New Virtual Machine Wizard* untuk menentukan jenis konfigurasi *virtual machine* yang ingin dibuat, seperti terlihat pada gambar berikut:



Terdapat 2 pilihan jenis konfigurasi yang dapat dipilih yaitu *Typical (recommended)* dan *Custom (advanced)*. Jenis konfigurasi *Typical* disarankan untuk dipilih ketika ingin membuat virtual machine melalui beberapa tahapan dengan mudah. Sebaliknya jenis konfigurasi *Custom* akan memberikan pilihan pengaturan lanjutan seperti penentuan

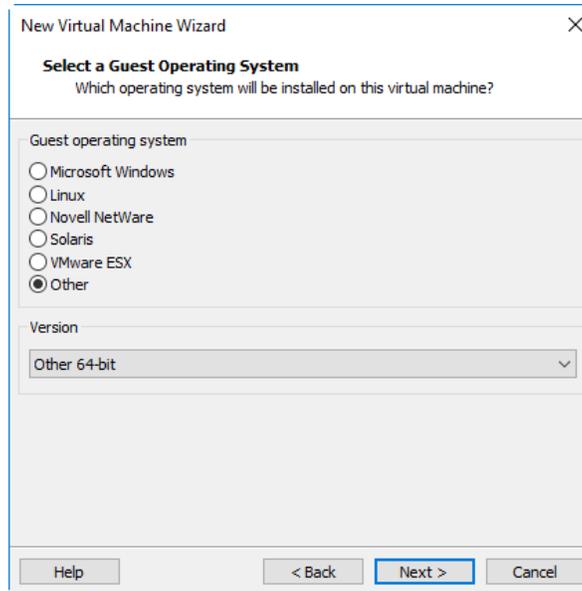
jenis *controller SCSI*, jenis *virtual disk* dan kompatibilitas dengan produk *VMWare* versi sebelumnya. Pilih **Typical**, dan klik tombol **Next >** untuk melanjutkan.

4. Tampil kotak dialog *Guest Operating System Installation* untuk menentukan bagaimana cara instalasi sistem operasi dilakukan, seperti terlihat pada gambar berikut:



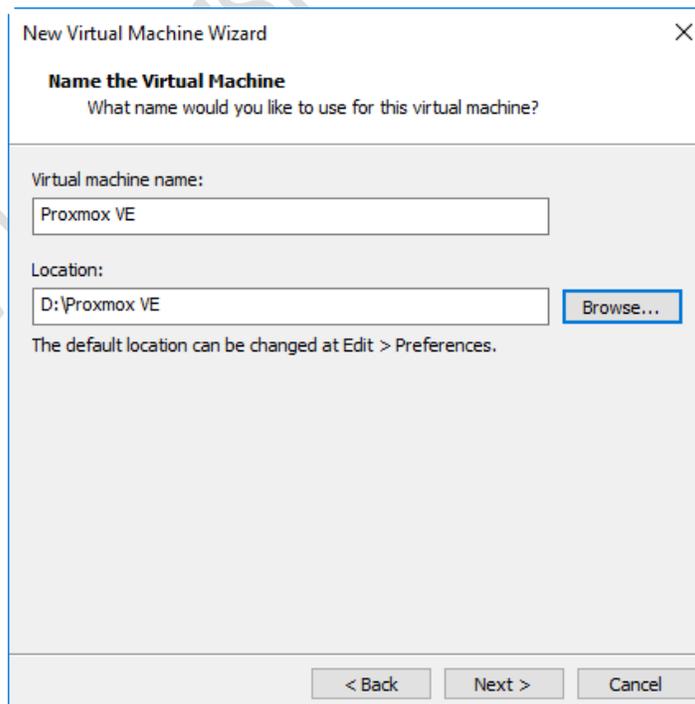
Terdapat 3 pilihan yaitu *Install from Installer disc* untuk menginstalasi dari media disc seperti CD/DVD, *Install from Installer disc image file (iso)* untuk menginstalasi dari file ISO, dan *I will install the operating system later* untuk mempersiapkan virtual machine dengan hardisk kosong tanpa melakukan instalasi sistem operasi. Pilih *I will install the operating system later*, dan klik tombol **Next >** untuk melanjutkan.

5. Tampil kotak dialog *Select a Guest Operating System* untuk menentukan jenis sistem operasi yang akan diinstalasi pada virtual machine yang dibuat, seperti ditunjukkan pada gambar berikut:



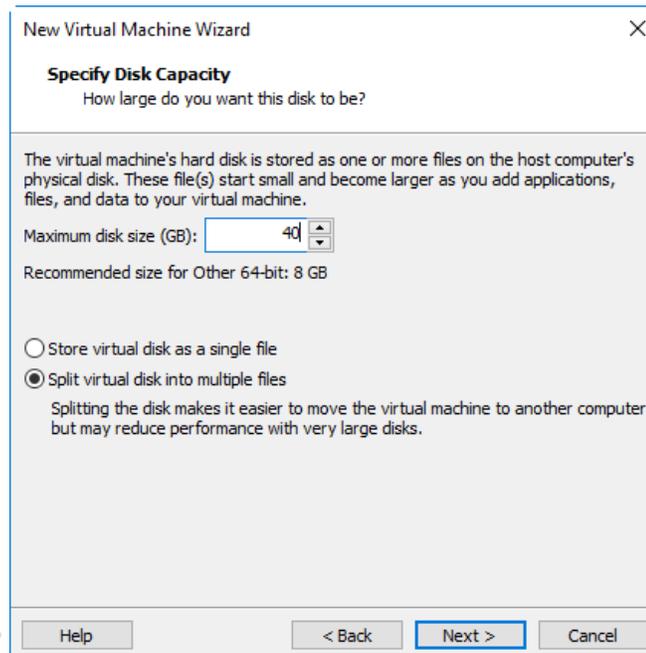
Pilih *Other* pada bagian *Guest operating system* dan *Other 64-bit* pada bagian *Version*. Klik tombol **Next >** untuk melanjutkan.

6. Tampil kotak dialog *Name the Virtual Machine* untuk menentukan nama pengenal *virtual machine* dan menentukan lokasi penyimpanan file *virtual machine* yang dibuat, seperti terlihat pada gambar berikut:



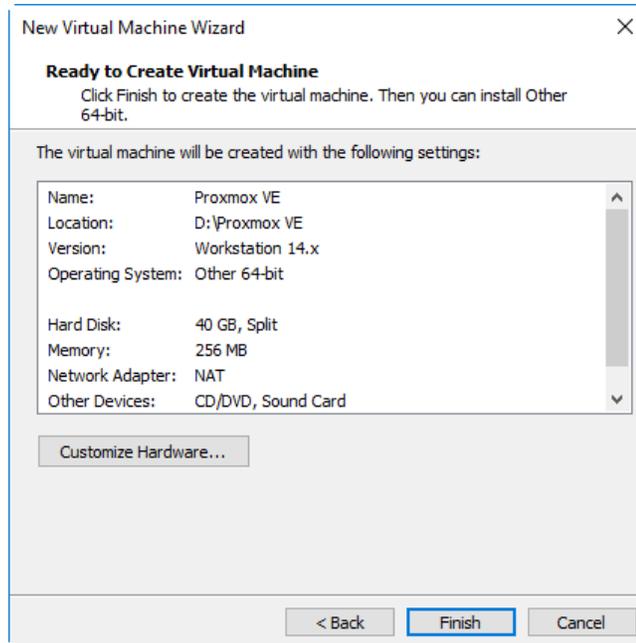
Pada bagian *Virtual machine name* masukkan nama pengenalan virtual machine, sebagai contoh **Proxmox VE**. Sedangkan pada bagian *Location* tentukan lokasi penyimpanan file virtual machine yang dibuat dengan cara menekan tombol *Browse ...* sebagai contoh diletakkan di **D:\Proxmox VE**. Klik tombol **Next >** untuk melanjutkan.

7. Tampil kotak dialog *Specify Disk Capacity* untuk menentukan kapasitas media penyimpanan yang dialokasikan untuk virtual machine yang dibuat, seperti terlihat pada gambar berikut:



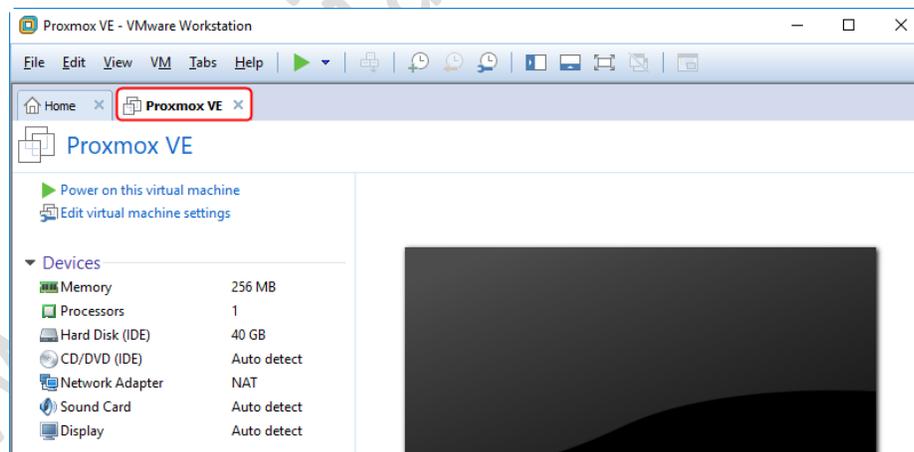
Pada bagian *Maximum disk size (GB)* masukkan kapasitas media penyimpanan (hardisk) yang dialokasikan untuk virtual machine yang dibuat, sebagai contoh 40 GB. Klik tombol **Next >** untuk melanjutkan.

8. Tampil kotak dialog *Ready to Create Virtual Machine* yang menampilkan informasi ringkasan pengaturan yang telah ditentukan untuk virtual machine yang akan dibuat, seperti terlihat pada gambar berikut:

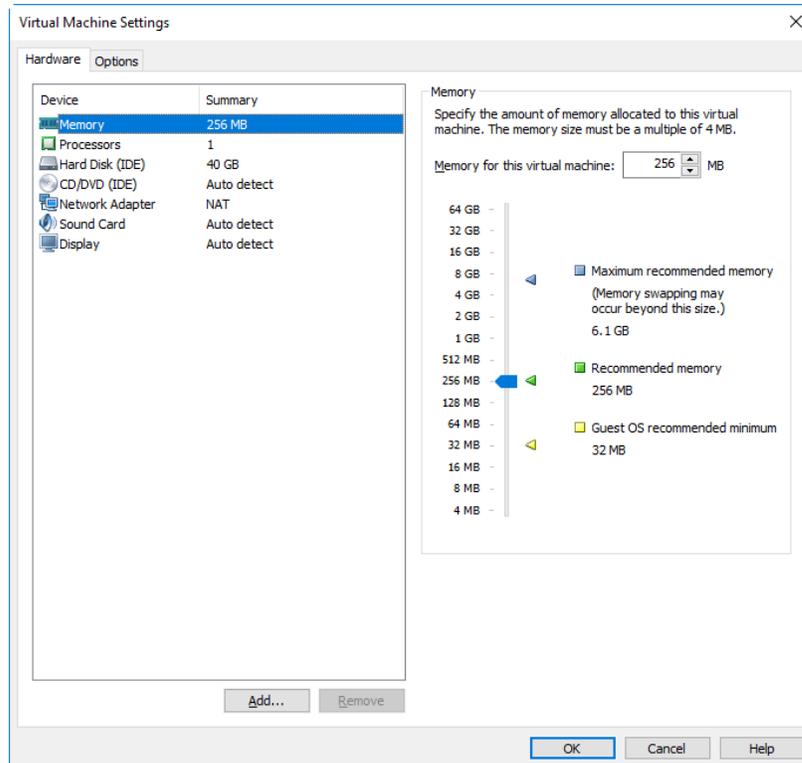


Klik tombol **Finish** untuk membuat virtual machine.

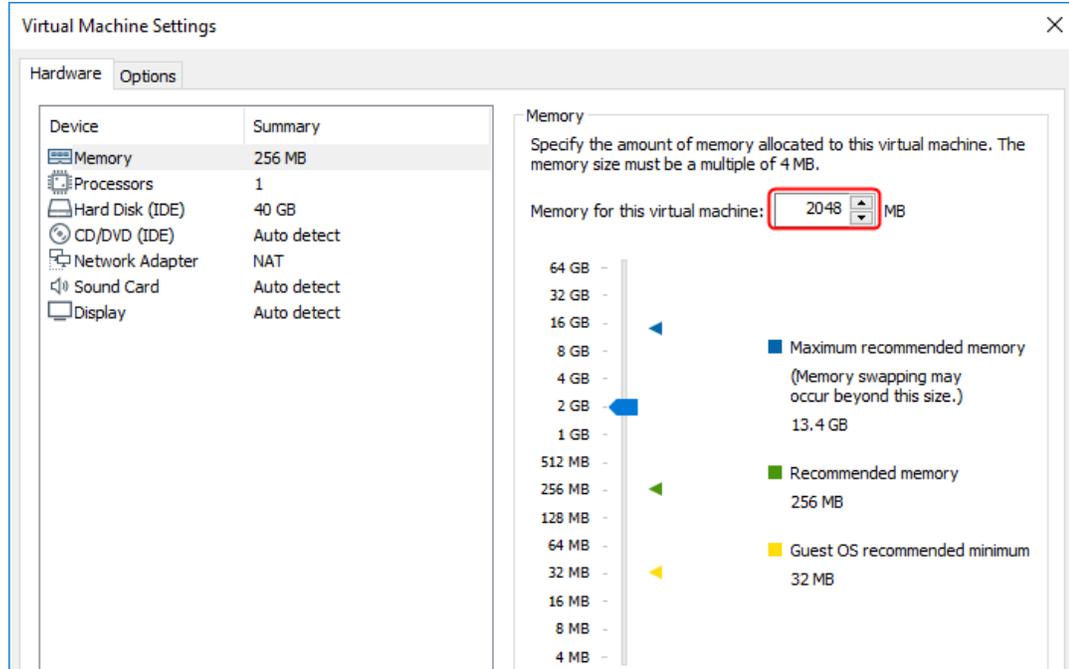
9. Tampil kotak dialog yang menampilkan *virtual machine* yang telah berhasil dibuat yaitu dengan nama pengenal **Proxmox VE**, seperti terlihat pada gambar berikut:



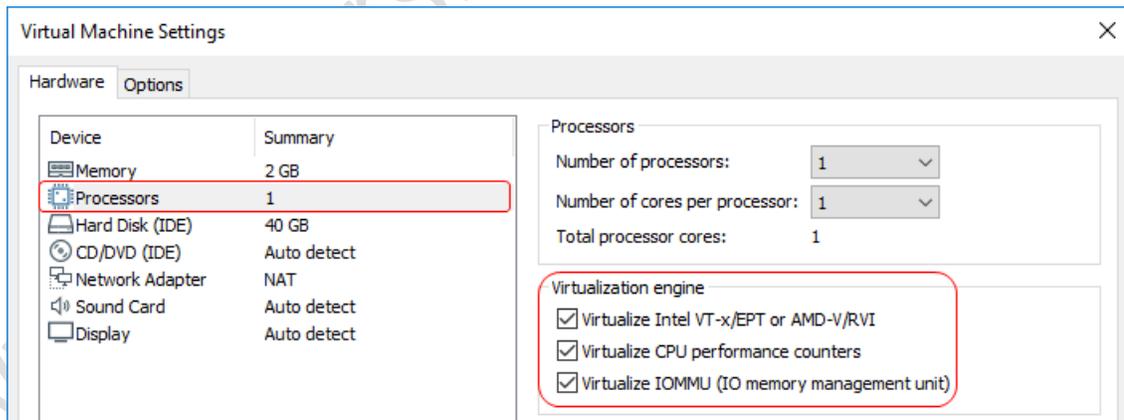
Selanjutnya klik *Edit virtual machine settings* untuk melakukan perubahan pada pengaturan *virtual machine* untuk beberapa komponen hardware, seperti terlihat pada gambar berikut:



10. Tampil kotak dialog *Virtual Machine Settings*. Pada tab *Hardware* di panel sebelah kiri pilih *Memory*. Selanjutnya pada panel detail sebelah kanan lakukan penyesuaian ukuran memori yang dialokasikan untuk *virtual machine* di parameter *Memory for this virtual Machine* sebagai contoh dialokasikan 2 GB, seperti terlihat pada gambar berikut:

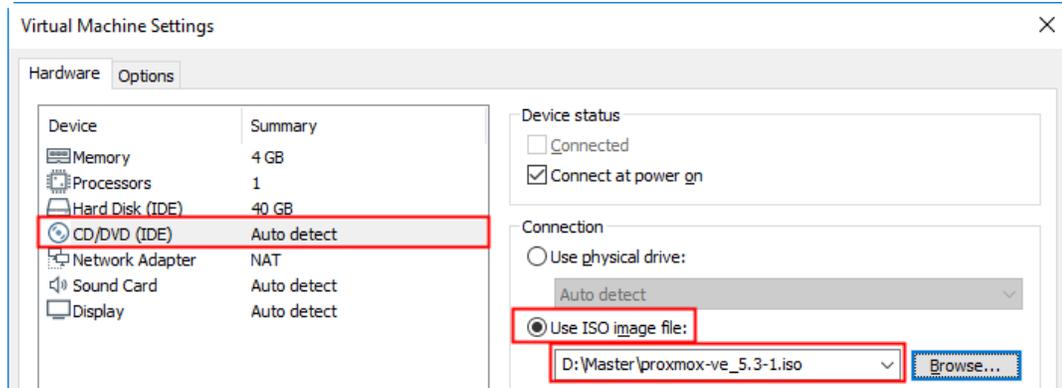


11. Pada tab *Hardware* di panel sebelah kiri dari *Virtual Machine Settings* pilih *Processors*. Selanjutnya pada panel sebelah kanan, lakukan pengaktifan *nested virtualization* atau *virtualization engine* dengan mencentang pada 3 (tiga) pilihan parameter yaitu antara lain *Virtualize Intel VT-x/EPT or AMD-V/RVI*, *Virtualize CPU performance counters* dan *Virtualize IOMMU (IO memory management unit)*, seperti terlihat pada gambar berikut:

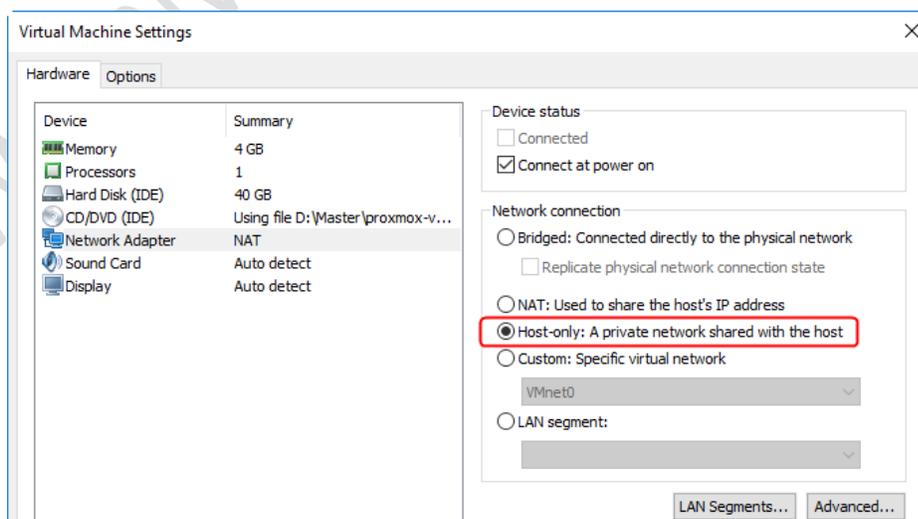


12. Pada tab *Hardware* di panel sebelah kiri dari *Virtual Machine Settings* pilih *CD/DVD (IDE)* untuk mengarahkan ke lokasi penyimpanan file ISO dari **Proxmox VE 5.3**. Selanjutnya pada panel sebelah kanan akan muncul detail pengaturan CD/DVD. Pada bagian *Connection*, pilih *Use ISO image file*, dan klik tombol *Browse...* untuk mengarahkan ke

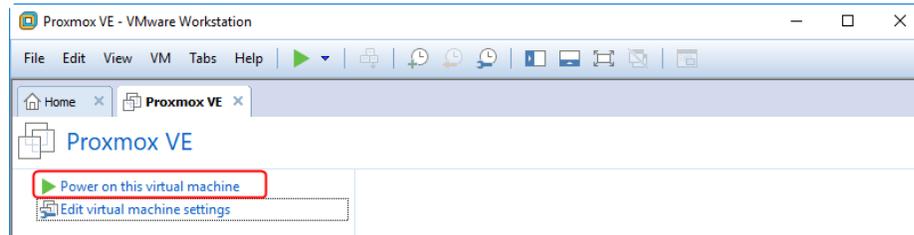
lokasi penyimpanan file ISO dari *Proxmox VE 5.3* yang akan digunakan sebagai media sumber instalasi, sebagai contoh terdapat di drive **D:\Master\proxmox-ve_5.3-1.iso**, seperti terlihat pada gambar berikut:



13. Pada tab *Hardware* di panel sebelah kiri dari *Virtual Machine Settings* pilih *Network Adapter*. Selanjutnya pada panel sebelah kanan akan muncul detail pengaturan *Network Adapter*. Pada bagian *Network connection* beberapa pilihan jenis koneksi jaringan yang dapat digunakan oleh *Network Adapter* yaitu *Bridged* (untuk dapat terhubung secara langsung ke jaringan fisik), *Network Address Translation (NAT)* - untuk berbagi pakai alamat IP dari host), *Host-only* (untuk terhubung ke jaringan privat yang dibagi pakai dengan host), dan *Custom* (untuk secara spesifik menentukan virtual network yang ingin digunakan). Pada bagian *Network connection* dipilih ***Host-only***, seperti terlihat pada gambar berikut:



Klik tombol **OK** untuk menutup kotak dialog *Virtual Machine Settings*. Selanjutnya klik **Power on the virtual machine** untuk menghidupkan *virtual machine* dan memulai instalasi *Proxmox VE 5.3* pada *virtual machine* yang telah dibuat, seperti terlihat pada gambar berikut:



14. Tampil menu awal instalasi berupa *Welcome to Proxmox Virtual Environment* untuk menentukan jenis instalasi yang akan dilakukan, seperti terlihat pada gambar berikut:

Proxmox VE 5.3 (iso release 1) - <http://www.proxmox.com/>



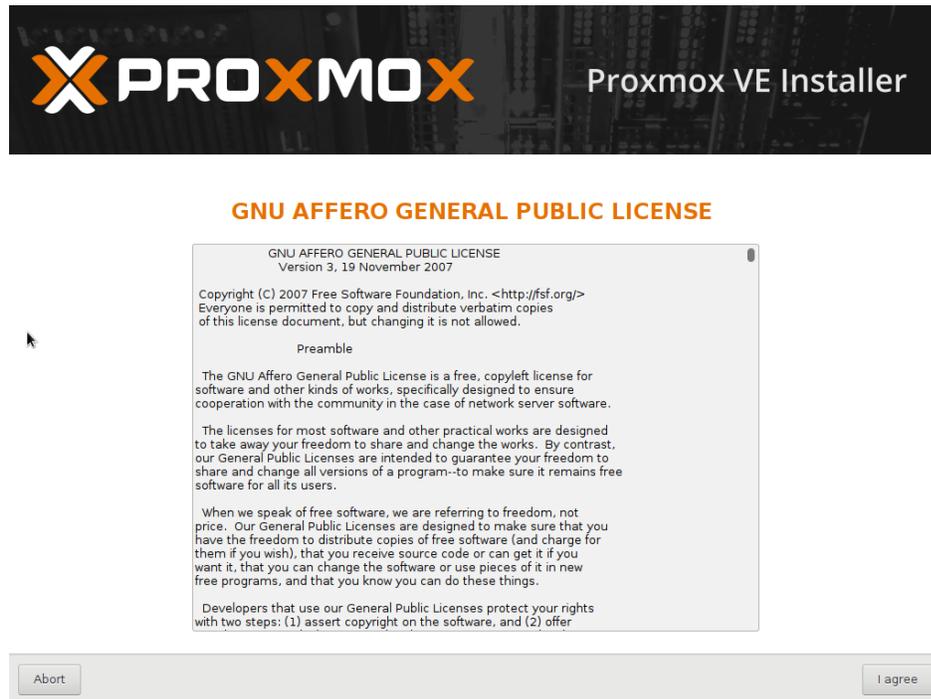
Welcome to Proxmox Virtual Environment

```

Install Proxmox VE
Install Proxmox VE (Debug mode)
Rescue Boot
Test memory
  
```

Terdapat beberapa pilihan yang tampil yaitu *Install Proxmox VE* (untuk menginstalasi secara normal), *Install Proxmox VE (Debug mode)* untuk menginstalasi pada mode debug yang akan membuka shell console pada beberapa tahapan instalasi dimana umumnya digunakan oleh developer, *Rescue Boot* (untuk memperbaiki sistem Proxmox yang telah terinstalasi ketika tidak dapat melakukan *booting* dengan normal), *Test Memory* (untuk melakukan pengujian pada RAM yang terpasang pada komputer apakah berfungsi dan bebas dari kesalahan atau *error*). Secara default telah terpilih **Install Proxmox VE**. Tekan tombol **Enter** untuk melanjutkan instalasi pada mode tersebut.

15. Tampil kotak dialog persetujuan lisensi “**GNU Affero General Public License**”, seperti terlihat pada gambar berikut:



Klik tombol **I Agree** untuk menyetujui lisensi dan melanjutkan instalasi.

16. Tampil kotak dialog **Proxmox Virtualization Environment (PVE)** untuk memilih **Target hardisk** sebagai lokasi instalasi, seperti terlihat pada gambar berikut:



Proxmox Virtualization Environment (PVE)

The Proxmox Installer automatically partitions your hard disk. It installs all required packages and finally makes the system bootable from hard disk. All existing partitions and data will be lost.

Press the Next button to continue installation.

- **Please verify the installation target**
The displayed hard disk is used for installation. Warning: All existing partitions and data will be lost.
- **Automatic hardware detection**
The installer automatically configures your hardware.
- **Graphical user interface**
Final configuration will be done on the graphical user interface via a web browser.



Terlihat **Target Harddisk** yang telah terpilih adalah **/dev/sda** dengan kapasitas **40GB**. *Installer Proxmox* akan secara otomatis membuat partisi pada hardisk dan menginstalasi paket-paket yang dibutuhkan serta membuat sistem dapat di boot dari hardisk. **Perhatian:** keseluruhan partisi dan data akan hilang. *Installer* akan menggunakan *Logical Volume Manager (LVM)* apabila file system yang dipilih adalah **ext3**, **ext4** atau **xfs**. Secara default telah terpilih **ext4**. Jika diperlukan dapat pula dilakukan pengaturan jenis *file system* dan parameter LVM lainnya dengan menekan tombol **Options**.

Klik tombol **Next** untuk melanjutkan instalasi.

17. Tampil kotak dialog “**Location and Time Zone selection**” untuk mengatur *Country*, *Time zone* dan *Keyboard Layout*, seperti terlihat pada gambar berikut:



Location and Time Zone selection

The **Proxmox Installer** automatically makes location based optimizations, like choosing the nearest mirror to download files. Also make sure to select the right time zone and keyboard layout.

Press the Next button to continue installation.

- **Country:** The selected country is used to choose nearby mirror servers. This will speedup downloads and make updates more reliable.
- **Time Zone:** Automatically adjust daylight saving time.
- **Keyboard Layout:** Choose your keyboard layout.

 The screenshot shows a configuration window with three fields: 'Country' with a text input containing 'Indonesia', 'Time zone' with a dropdown menu showing 'Asia/Makassar', and 'Keyboard Layout' with a dropdown menu showing 'U.S. English'. At the bottom left is an 'Abort' button and at the bottom right is a 'Next' button.

Pada isian **Country** masukkan **Indonesia**. Sedangkan pengaturan zone waktu dapat dilakukan dengan memilih menu *dropdown* dari parameter **Time zone**. Untuk Waktu Indonesia Barat (WIB) pilih *Asia/Jakarta*, untuk Waktu Indonesia Tengah (WITA) pilih *Asia/Makassar*, sedangkan Wilayah Indonesia Timur (WIT) pilih *Asia/Jayapura*. Pilih **Asia/Makassar**. Klik tombol **Next** untuk melanjutkan.

18. Tampil kotak dialog **Administration Password and E-mail Address** untuk mengatur *Password* dari user "*root*" dan *E-mail*, seperti terlihat pada gambar berikut:



Administration Password and E-Mail Address

Proxmox Virtual Environment is a full featured highly secure GNU/Linux system based on Debian.

Please provide the root password in this step.

- **Password:** Please use a strong password. It should have 8 or more characters. Also combine letters, numbers, and symbols.

- **E-Mail:** Enter a valid email address. Your Proxmox VE server will send important alert notifications to this email account (such as backup failures, high availability events, etc.).

Press the Next button to continue installation.

 The screenshot shows a configuration window with three input fields: 'Password' (masked with dots), 'Confirm' (masked with dots), and 'E-Mail' (containing 'admin@iputuhariyadi.net'). There are 'Abort' and 'Next' buttons at the bottom.

Pada isian **Password** dan **Confirm**, masukkan sandi login yang akan digunakan oleh user “root”, sebagai contoh “12345678”. Sedangkan pada isian **E-mail**, masukkan alamat untuk yang akan digunakan oleh Proxmox untuk mengirimkan notifikasi terkait kegagalan *backup, high availability events*, dan lainnya, sebagai contoh **admin@iputuhariyadi.net**. Tekan tombol **Next** untuk melanjutkan instalasi.

19. Tampil kotak dialog **Management Network Configuration** untuk mengatur konfigurasi jaringan. Lengkapi isian dari masing-masing parameter berikut:

- a) **Hostname (FQDN)**, masukkan nama komputer dengan format *Fully Qualified Domain Name*, sebagai contoh **pve.iputuhariyadi.net**.
- b) **IP Address**, masukkan alamat IP yang digunakan oleh Proxmox yaitu **192.168.169.1** sesuai dengan rancangan jaringan ujicoba.
- c) **Netmask**, masukkan alamat subnetmask yaitu 255.255.255.0.
- d) **Gateway**, masukkan alamat *gateway* untuk komunikasi ke beda jaringan atau ke *Internet*, sebagai contoh **192.168.169.254**.
- e) **DNS Server**, masukkan alamat *server Domain Name System (DNS)* untuk mentranslasikan nama domain ke alamat IP dan sebaliknya, sebagai contoh **192.168.169.254**.

seperti terlihat pada gambar berikut:



Management Network Configuration

Please verify the displayed network configuration. You will need a valid network configuration to access the management interface after installation.

Afterwards press the Next button to continue installation. The installer will then partition your hard disk and start copying packages.

- **IP address:** Set the IP address for the Proxmox Virtual Environment.
- **Netmask:** Set the netmask of your network.
- **Gateway:** IP address of your gateway or firewall.
- **DNS Server:** IP address of your DNS server.

Management Interface: ens32 - 00:0c:29:3a:a1:88 (e1000) ▼

Hostname (FQDN): pve.iputiharyadi.net

IP Address: 192.168.169.1

Netmask: 255.255.255.0

Gateway: 192.168.169.254

DNS Server: 192.168.169.254

Abort Next

Tekan tombol **Next** untuk melanjutkan instalasi.

20. Tampil kotak dialog yang menampilkan proses pembuatan partisi, format hardisk dan penyalinan paket-paket ke target hardisk, seperti terlihat pada gambar berikut:



Virtualization Platform

Open Source Virtualization Platform

- Enterprise ready
- Central Management
- Clustering
- Online Backup solution
- Live Migration
- 32 and 64 bit guests

Visit www.proxmox.com for additional information and the Wiki about Proxmox VE.

- **Container Virtualization**
Only 1-3% performance loss using OS virtualization as compared to using a standalone server.
- **Full Virtualization (KVM)**
Run unmodified virtual servers - Linux or Windows.

create partitions 0%

Abort Next



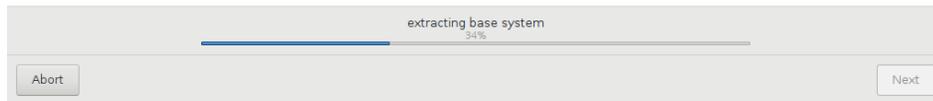
Virtualize your IT Infrastructure

Proxmox VE is ready for enterprise deployments.

The role based permission management combined with the integration of multiple external authentication sources is the base for a secure and stable environment.

Visit www.proxmox.com for more information about commercial support subscriptions.

- **Commitment to Free Software**
The source code is released under the GNU Affero General Public License.
- **RESTful web API**
Resource Oriented Architecture (ROA) and declarative API definition using JSON Schema enables easy integration for third party management tools.
- **Virtual Appliances**
Pre-installed applications - up and running within a few seconds.



Tunggu hingga proses instalasi selesai.

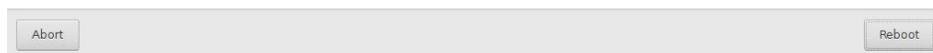
21. Tampil kotak dialog **Installation successful!** yang menginformasikan instalasi *Proxmox VE* telah selesai diinstalasi dan siap digunakan, seperti terlihat pada gambar berikut:



Installation successful!

The Proxmox Virtual Environment is now installed and ready to use.

- **Next steps**
Reboot and point your web browser to the selected IP address.
Also visit www.proxmox.com for more information.



Tekan tombol **Reboot**. Tunggu hingga proses *reboot* selesai dilakukan. Setelah proses *reboot* selesai dilakukan maka akan tampak prompt login untuk otentikasi sebelum dapat mengakses sistem *Proxmox*, seperti terlihat pada gambar berikut:

```
-----
Welcome to the Proxmox Virtual Environment. Please use your web browser to
configure this server - connect to:

https://192.168.169.1:8006/

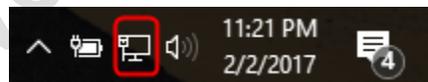
-----
pve login: _
```

Konfigurasi selanjutnya dapat dilakukan melalui antarmuka web dari *Proxmox* yang dapat diakses pada alamat **http://192.168.169.1:8006**.

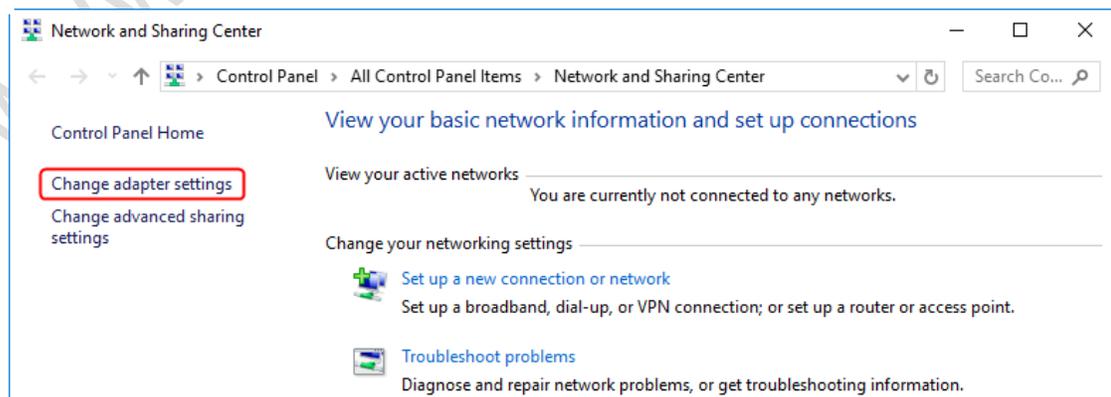
C. Konfigurasi Client Windows 10

Adapun langkah-langkah konfigurasi yang dilakukan pada *Client Windows 10* adalah sebagai berikut:

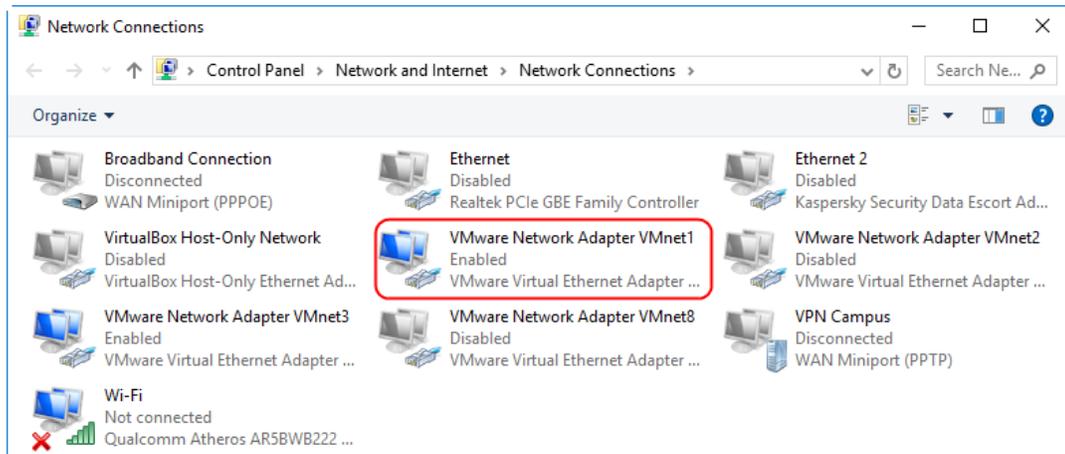
1. Mengatur pengalamatan IP dan parameter TCP/IP lainnya melalui **taskbar bagian pojok kanan bawah** dengan cara **klik kanan** pada icon **Network** dan pilih **Open Network & Sharing Center**, seperti terlihat pada gambar berikut:



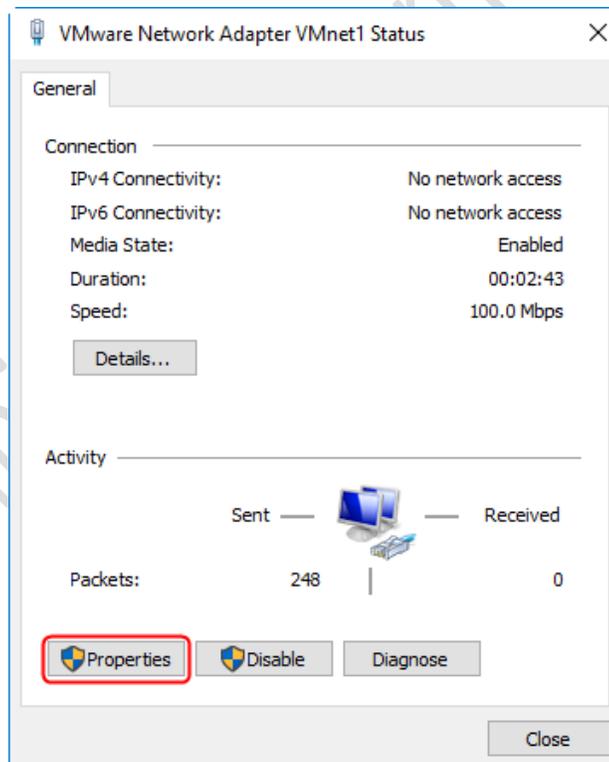
2. Tampil kotak dialog **Network and Sharing Center**. Pilih **Change Adapter Settings**, seperti terlihat pada gambar berikut:



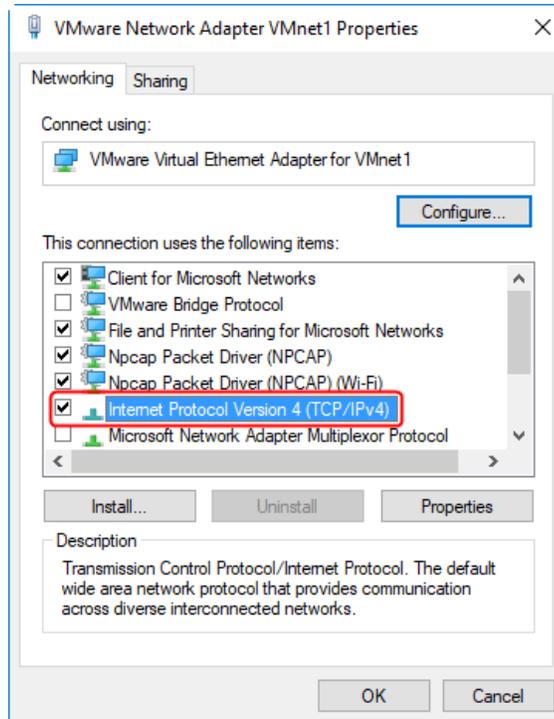
3. Tampil kotak dialog **Network Connections**. Klik dua kali pada **VMWare Network Adapter VMnet1**, seperti terlihat pada gambar berikut:



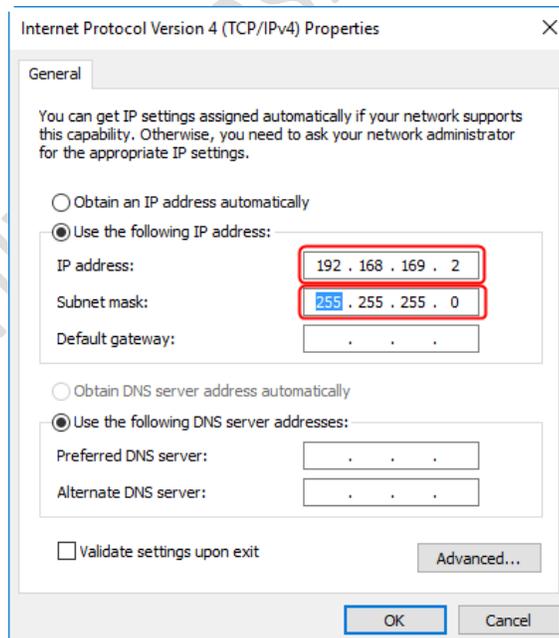
4. Tampil kotak dialog **VMware Network Adapter VMnet1 Status**. Klik tombol **Properties**, seperti terlihat pada gambar berikut:



5. Tampil kotak dialog **VMware Network Adapter VMnet1 Properties**. Pada bagian **“This connection uses the following items:”**, klik dua kali pada pilihan **Internet Protocol Version 4 (TCP/IPv4)**, seperti terlihat pada gambar berikut:

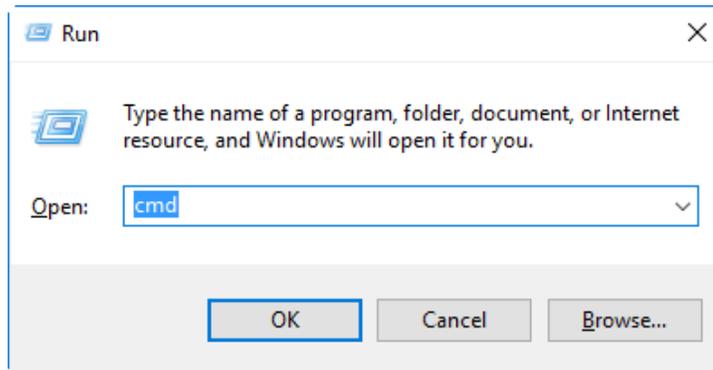


6. Tampil kotak dialog **Internet Protocol Version 4 (TCP/IPv4) Properties**. Pilih *Use the following IP Address*, seperti terlihat pada gambar berikut:

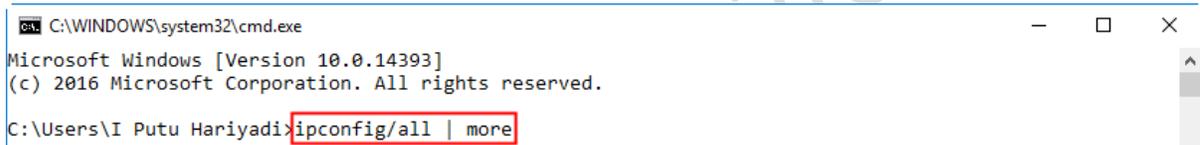


Pada isian **IP address**;, masukkan **192.168.169.2**. Sedangkan pada isian **Subnet mask**;, masukkan **255.255.255.0**. Klik tombol **OK > OK > Close**. Tutup kotak dialog **Network and Sharing Center**.

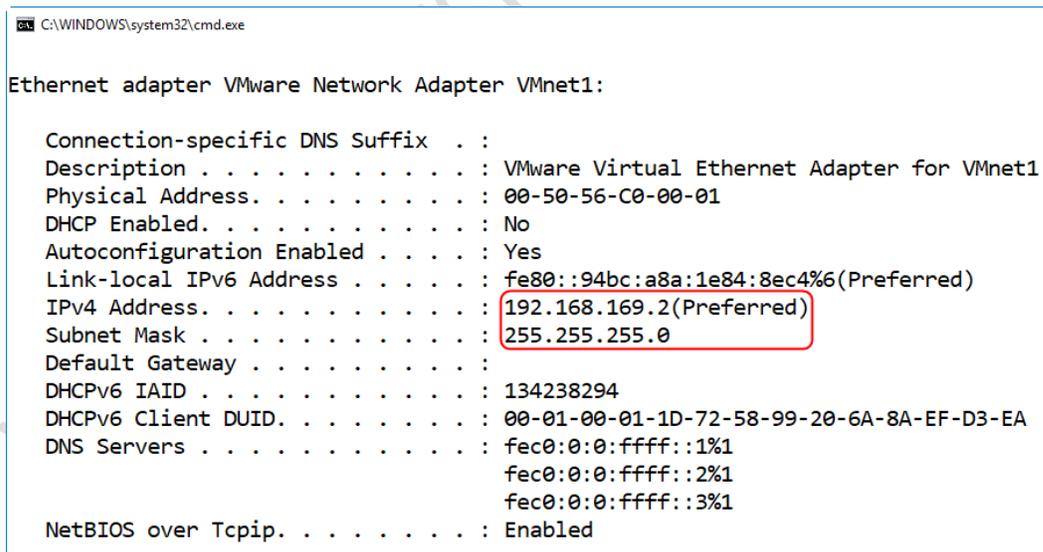
7. Buka **Command Prompt Windows** dengan menekan tombol **Windows+R**. Pada inputan form yang tampil, ketik **cmd** dan tekan tombol **Enter**.



8. Pada **Command Prompt** masukkan perintah **ipconfig/all | more** untuk memverifikasi pengalamatan IP yang telah diatur, seperti terlihat pada gambar berikut:



Tekan tombol **spasi** untuk menampilkan layar berikutnya. Pastikan adapter **VMware Network Adapter VMnet1** telah menggunakan alamat IP dan subnetmask yang telah diatur pada langkah sebelumnya, seperti terlihat pada gambar berikut:



Tekan tombol **q** untuk keluar.

9. Verifikasi koneksi dari *client Windows 10* ke *VM Proxmox* menggunakan perintah “**ping 192.168.169.1**” pada **Command Prompt Windows**, seperti terlihat pada gambar berikut:

```
C:\WINDOWS\system32\cmd.exe

C:\Users\I Putu Hariyadi>ping 192.168.169.1

Pinging 192.168.169.1 with 32 bytes of data:
Reply from 192.168.169.1: bytes=32 time<1ms TTL=64

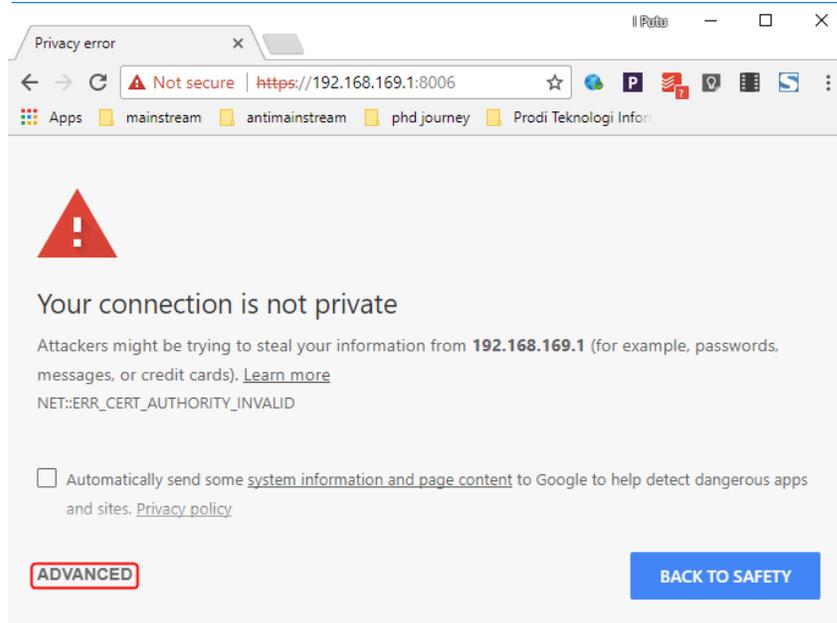
Ping statistics for 192.168.169.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Terlihat koneksi ke *VM Proxmox* telah berhasil dilakukan.

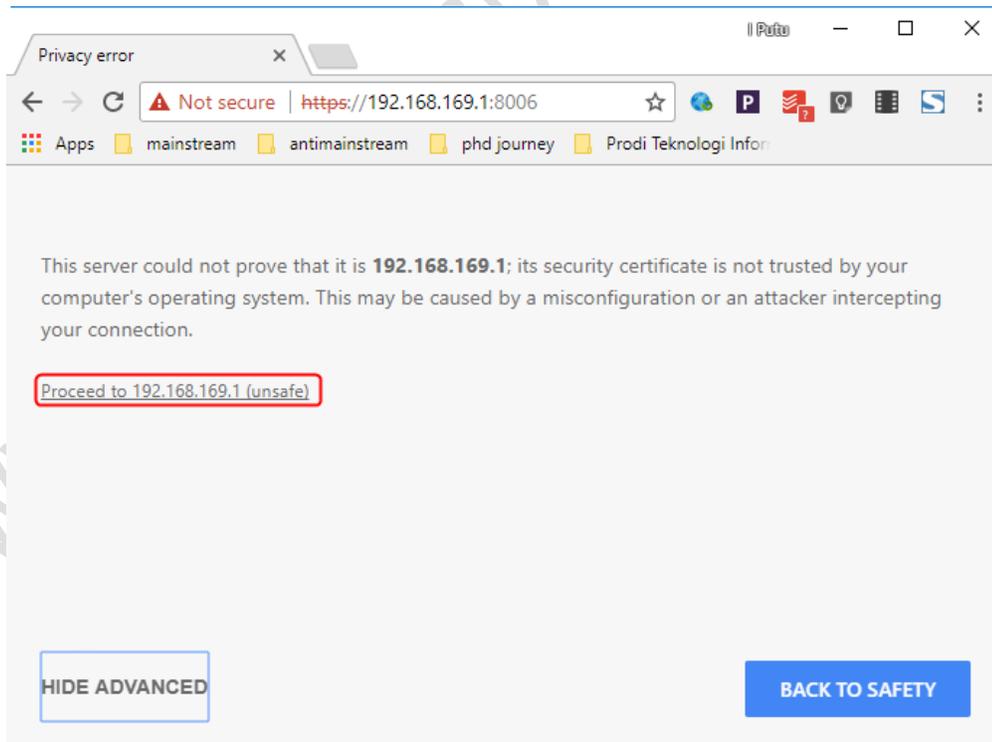
D. Konfigurasi Proxmox VE 5.3

Adapun langkah-langkah untuk mengkonfigurasi Proxmox VE 5.3 melalui antarmuka berbasis web adalah sebagai berikut:

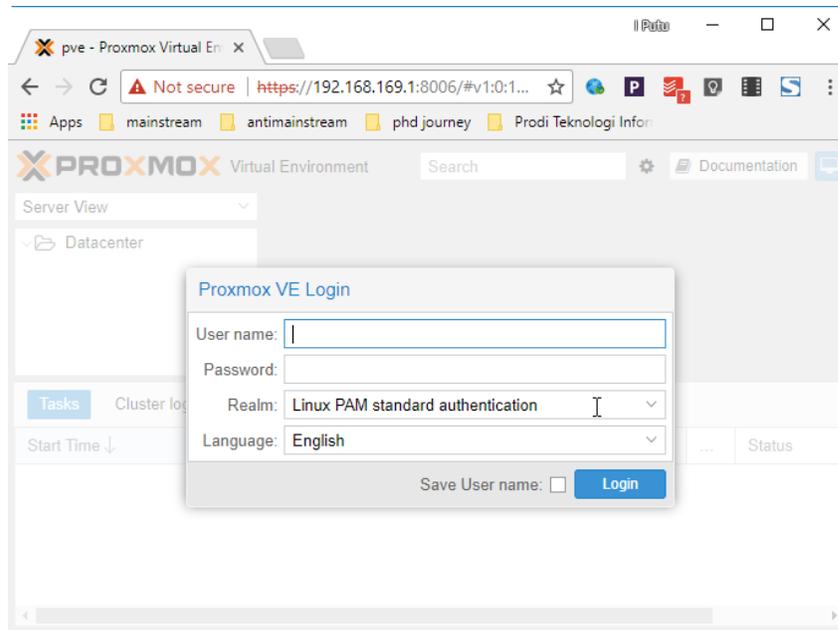
1. Buka *browser* sebagai contoh menggunakan **Chrome**. Pada address bar dari browser, masukkan URL <https://192.168.169.1:8006>. Hasil pengaksesan, seperti terlihat pada gambar berikut:



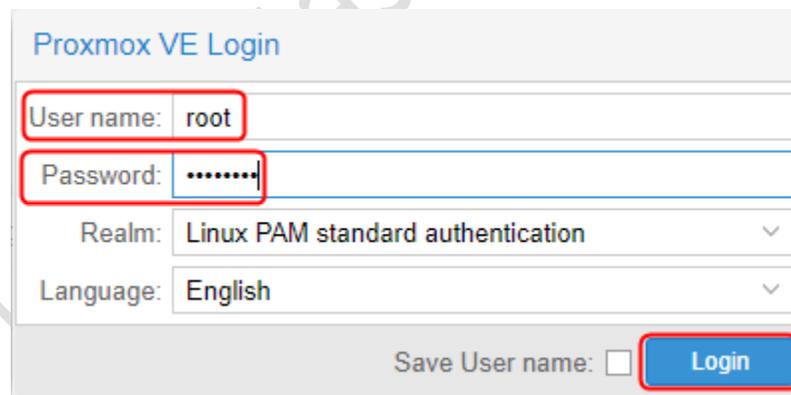
Tampil pesan peringatan **“Your connection is not private”**. Klik **Advanced** untuk melanjutkan pengaksesan dan klik link **“Proceed to 192.168.169.1 (unsafe)”**, seperti terlihat pada gambar berikut:



Maka *web interface* dari konfigurasi *Proxmox* berhasil diakses, seperti terlihat pada gambar berikut:

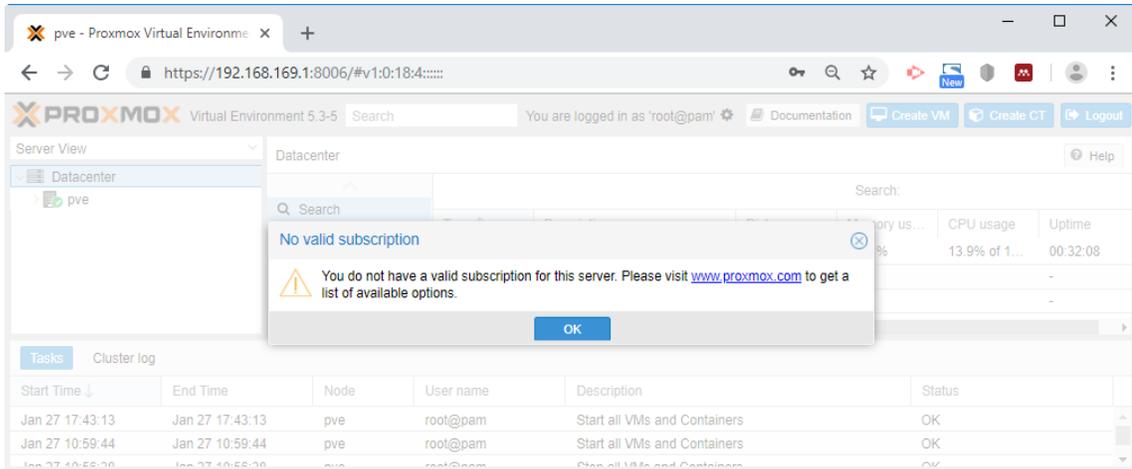


2. Pada kotak dialog otentikasi *Proxmox VE Login*, lengkapi isian “**User name**” dan “**Password**”. Pada isian “*User name*”, masukkan “**root**”. Sedangkan pada isian “*Password*”, masukkan sandi login dari user “*root*” yaitu **12345678**, seperti terlihat pada gambar berikut:



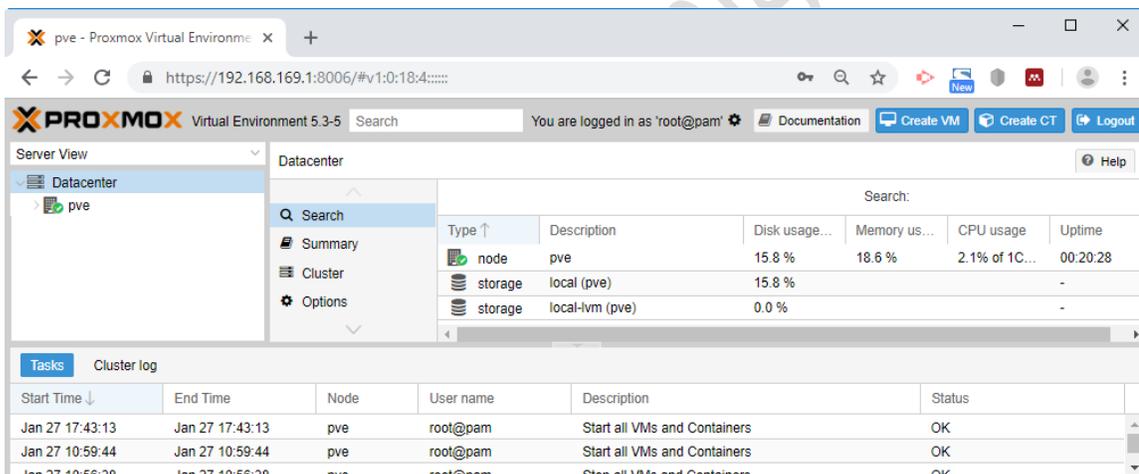
Klik tombol **Login**.

3. Tampil kotak dialog “**No valid subscription**” yang menginformasikan bahwa Anda tidak memiliki *subscription* yang valid untuk server ini, seperti terlihat pada gambar berikut:



Pilihan jenis *subscription* dapat diakses lebih lanjut pada situs Proxmox di alamat www.proxmox.com. Klik tombol **OK**.

4. Tampil halaman *Server View* dari *Proxmox*, seperti terlihat pada gambar berikut:



Selanjutnya Anda dapat melakukan aktivitas manajemen Proxmox seperti mengunggah file *image Template Linux Container* atau *file ISO image*, pembuatan *Virtual Machine (Create VM)* atau *Container (Create CT)* dan lain sebagainya.

Untuk keluar dari *web interface* konfigurasi *Proxmox*, klik **Logout**.

BAB II

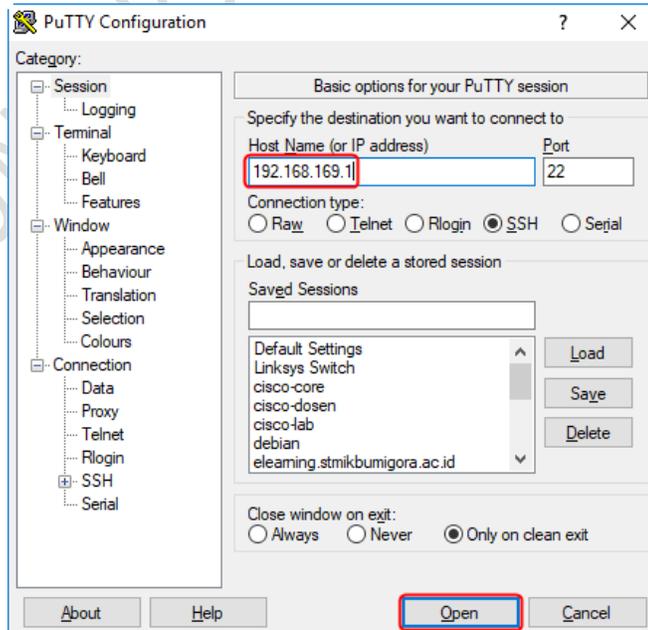
MENONAKTIFKAN PESAN NOTIFIKASI “NO VALID SUBSCRIPTION”

PADA PROXMOX VE 5.3

Setelah berhasil melakukan “Instalasi dan Konfigurasi Proxmox VE 5.3 pada VMware Workstation 14” di bab sebelumnya maka selanjutnya akan dilakukan penonaktifan pesan notifikasi “No Valid Subscription”. Pesan notifikasi ini akan selalu tampil ketika pengguna telah berhasil melalui proses otentikasi login pada *web interface Proxmox*. Proses penonaktifan pesan notifikasi tersebut memerlukan akses *console* atau *remote access* melalui *Secure Shell (SSH)*.

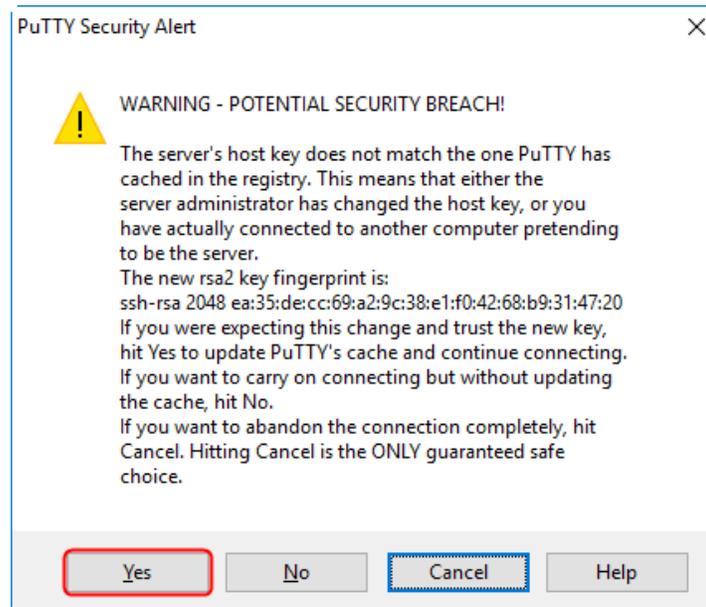
Adapun langkah-langkah penonaktifan pesan notifikasi tersebut melalui *SSH* adalah sebagai berikut:

1. Jalankan aplikasi *SSH Client*, sebagai contoh menggunakan *PuTTY*. Tampil kotak dialog *PuTTY Configuration*. Pada isian **Host Name (or IP Address)**, masukkan alamat IP dari *Server Proxmox* yaitu **192.168.169.1**, seperti terlihat pada gambar berikut:



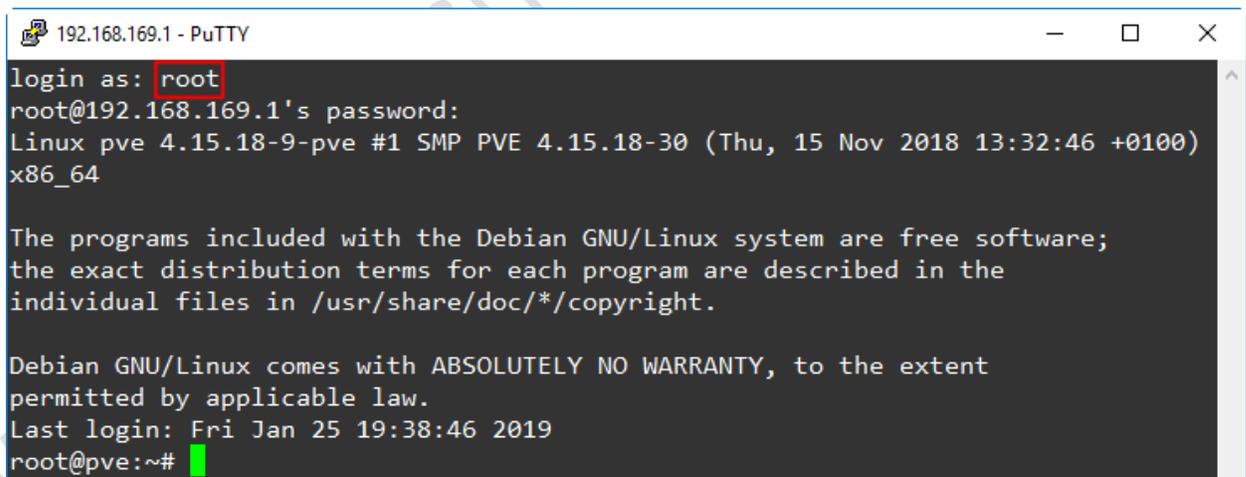
Klik tombol **Open**.

2. Tampil kotak dialog **PuTTY Security Alert** yang menampilkan pesan peringatan terkait potensi pelanggaran keamanan, seperti terlihat pada gambar berikut:



Klik tombol **Yes** untuk melanjutkan.

3. Tampil kotak dialog *PuTTY* yang meminta pengguna untuk melakukan proses otentikasi login ke *Server Proxmox*, seperti terlihat pada gambar berikut:



Pada inputan **login as:**, masukkan **“root”** dan tekan tombol **Enter**. Selanjutnya tampil inputan **password:**, masukkan **“12345678”** dan tekan tombol **Enter**. Apabila proses otentikasi berhasil dilakukan maka akan tampil *shell prompt #*.

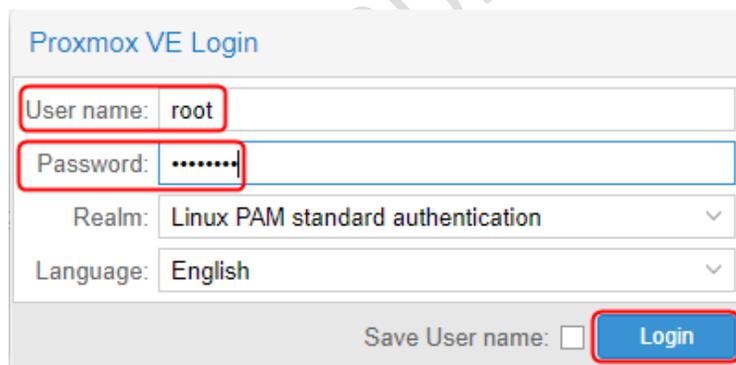
4. Mengubah parameter pada file `"/usr/share/javascript/proxmox-widget-toolkit/proxmoxlib.js"` yaitu: `if (data.status !== 'Active')` menjadi `if (false)` dengan cara mengeksekusi perintah berikut:

```
# sed -i "s/data.status !== 'Active'/false/g"
/usr/share/javascript/proxmox-widget-toolkit/proxmoxlib.js
```

5. Keluar dari SSH.

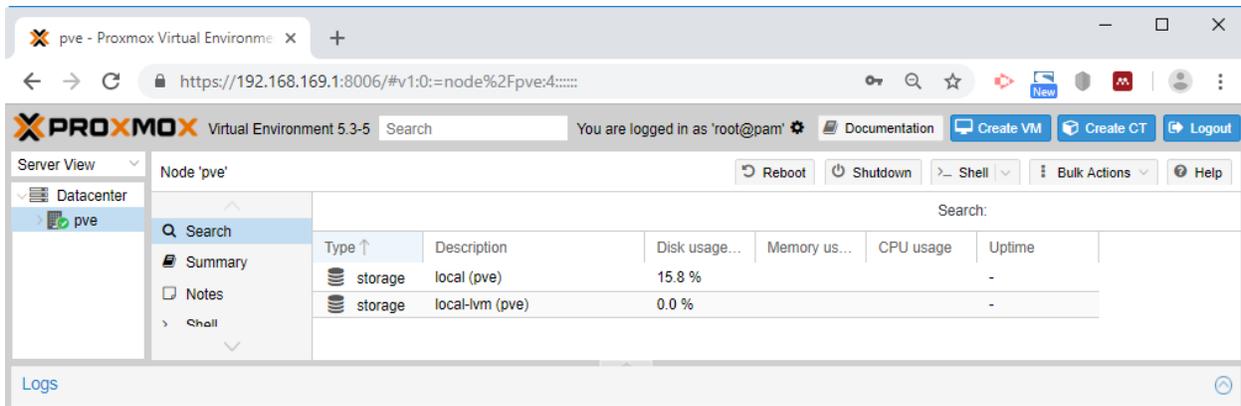
```
# exit
```

6. Memverifikasi hasil penonaktifkan pesan notifikasi dengan mengakses *web interface* dari *Proxmox*. Buka *browser*, sebagai contoh menggunakan **Chrome**. Pada *address bar* dari browser, masukkan URL <https://192.168.169.1:8006>.
7. Tampil kotak dialog otentikasi *Proxmox VE Login*, lengkapi isian **"User name"** dan **"Password"**. Pada isian **"User name"**, masukkan **"root"**. Sedangkan pada isian **"Password"**, masukkan sandi login dari user **"root"** yaitu **12345678**, seperti terlihat pada gambar berikut:



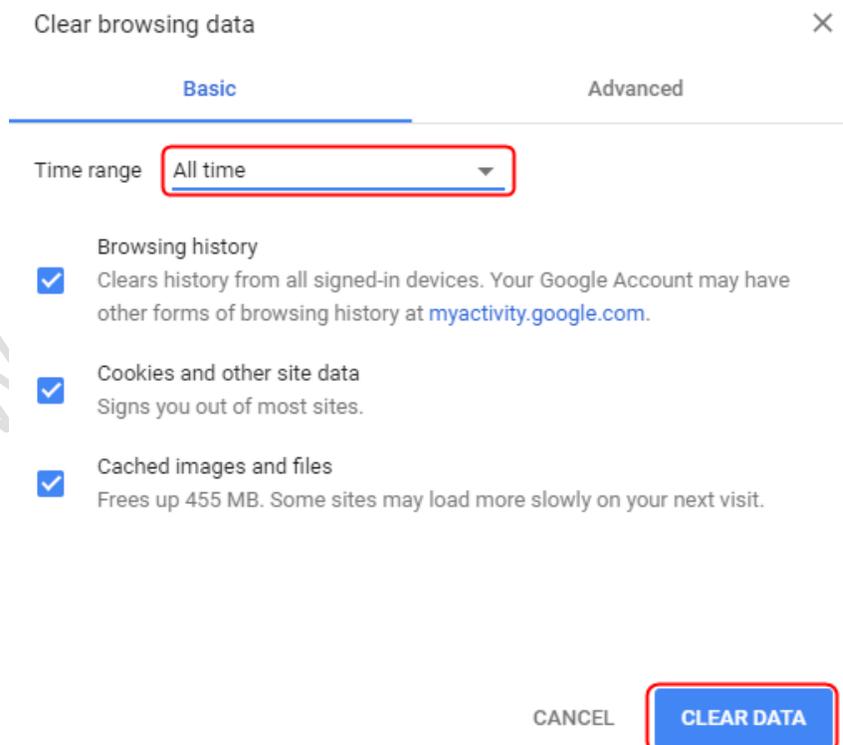
The image shows a 'Proxmox VE Login' dialog box. It has four input fields: 'User name' with the value 'root', 'Password' with masked characters '*****', 'Realm' with the value 'Linux PAM standard authentication', and 'Language' with the value 'English'. At the bottom right, there is a 'Login' button and a 'Save User name' checkbox which is unchecked.

Klik tombol **Login**. Pesan notifikasi **"No Valid Subscription"** tidak tampil. Pengguna langsung diarahkan ke tampilan halaman *Server View* dari *Proxmox*, seperti pada gambar berikut:



Selamat Anda telah berhasil menonaktifkan pesan notifikasi “**No Valid Subscription**” pada *Proxmox VE 5.3*. Untuk keluar dari *web interface* konfigurasi *Proxmox*, klik **Logout**.

Apabila pesan notifikasi masih tampil, maka lakukan penghapusan **Cookies** dari *browser* dengan menekan tombol **CTRL+SHIFT+DEL**. Tampil kotak dialog, **Clear browsing data**. Pada pilihan **Time range**, pilih **All time**, dan tekan tombol **CLEAR DATA**, seperti terlihat pada gambar berikut:



Selanjutnya lakukan percobaan pengaksesan kembali ke web interface dari Proxmox pada alamat **<https://192.168.169.1:8006>**.

WWW.UNIVERSITASBUMIGORA.AC.ID

BAB III

MENONAKTIFKAN PVE ENTERPRISE SUBSCRIPTION

DAN MENGAKTIFKAN PVE NO-SUBSCRIPTION REPOSITORY

PADA PROXMOX VE 5.3

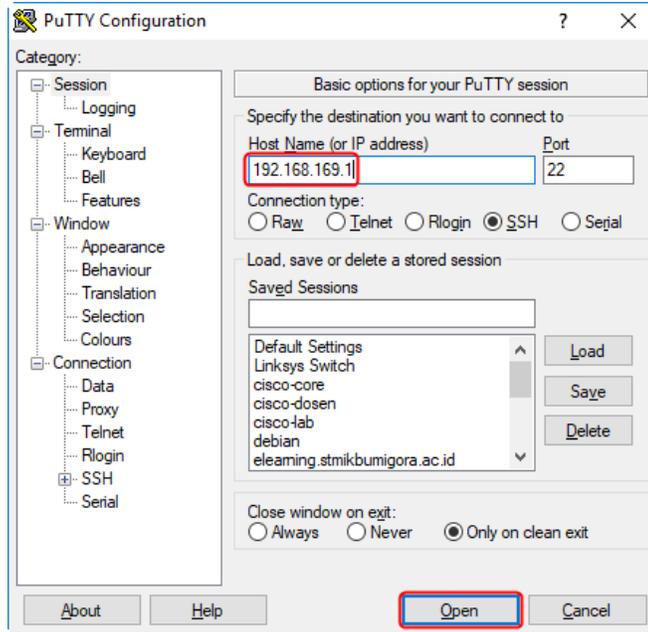
Menurut *Wiki* dari *Proxmox*, secara *default PVE Enterprise Subscription* telah aktif dan merupakan *repository default* dan direkomendasikan bagi pengguna PVE yang melakukan *subscription* karena memuat paket yang paling stabil sehingga sangat cocok digunakan untuk *production*. Untuk dapat memanfaatkan *repository* ini maka diperlukan *subscription key* yang **BERBAYAR**. Detail informasi pembiayaan terkait *PVE subscription* dapat dilihat pada alamat <https://www.proxmox.com/en/proxmox-ve/pricing>. File yang memuat pengaturan *PVE Enterprise Subscription* adalah `/etc/apt/sources.list.d/pve-enterprise.list`, dengan konten seperti berikut:

```
deb https://enterprise.proxmox.com/debian/pve stretch pve-enterprise
```

Penonaktifan *PVE Enterprise subscription* diperlukan apabila tidak memiliki *subscription key* sehingga tidak memunculkan pesan kesalahan. Bagi pengguna PVE yang tidak memiliki *subscription key* dapat memanfaatkan *PVE No-subscription repository*.

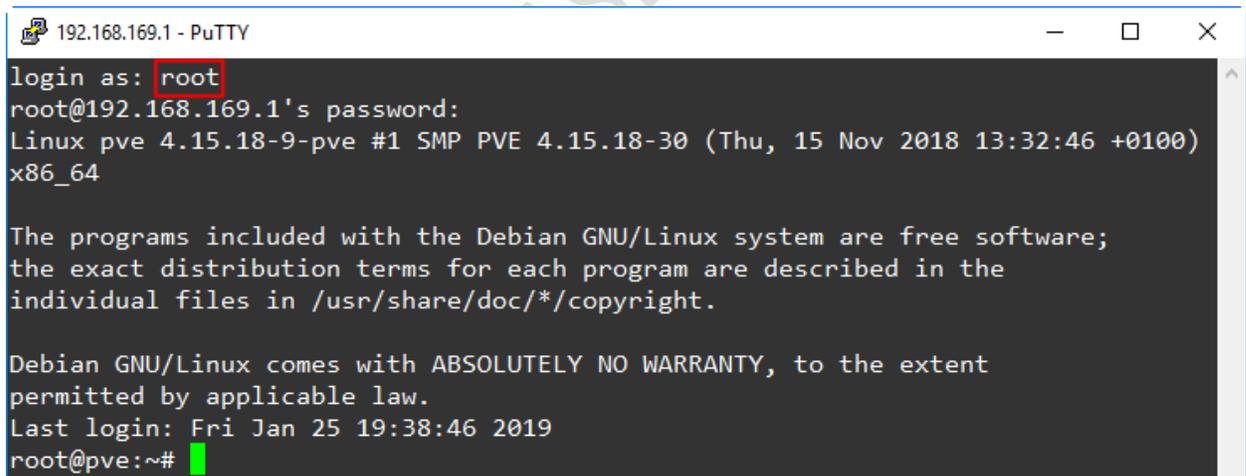
Adapun langkah-langkah untuk menonaktifkan *PVE Enterprise Subscription* dan mengaktifkan *PVE NO-subscription* melalui *SSH* adalah sebagai berikut:

1. Jalankan aplikasi *SSH Client*, sebagai contoh menggunakan *Putty*. Tampil kotak dialog *Putty Configuration*. Pada isian **Host Name (or IP Address)**, masukkan alamat IP dari *Server Proxmox* yaitu **192.168.169.1**, seperti terlihat pada gambar berikut:



Klik tombol **Open**.

2. Tampil kotak dialog *PuTTY* yang meminta pengguna untuk melakukan proses otentikasi login ke *Server Proxmox*, seperti terlihat pada gambar berikut:



Pada inputan **login as:**, masukkan “**root**” dan tekan tombol **Enter**. Selanjutnya tampil inputan **password:**, masukkan “**12345678**” dan tekan tombol **Enter**. Apabila proses otentikasi berhasil dilakukan maka akan tampil *shell prompt #*.

3. Menonaktifkan *PVE Enterprise subscription* dengan cara menambahkan tanda # diawal baris dari konten pada file `/etc/apt/sources.list.d/pve-enterprise.list` menggunakan editor *nano*.

```
# nano /etc/apt/sources.list.d/pve-enterprise.list
```

```
GNU nano 2.7.4 File: /etc/apt/sources.list.d/pve-enterprise.list
#deb https://enterprise.proxmox.com/debian/pve stretch pve-enterprise
```

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan **Enter**. Tekan tombol **CTRL+X** untuk keluar dari editor *nano*.

4. Memverifikasi hasil penonaktifan *PVE Enterprise subscription* menggunakan perintah “`cat /etc/apt/sources.list.d/pve-enterprise.list`”.

```
root@pve:~# cat /etc/apt/sources.list.d/pve-enterprise.list
#deb https://enterprise.proxmox.com/debian/pve stretch pve-enterprise
root@pve:~#
```

Terlihat pada awal baris dari file `pve-enterprise.list` telah terdapat tanda `#` yang berfungsi sebagai komentar.

5. Mengaktifkan *PVE No-subscription repository* dengan cara menambahkan parameter “`deb http://download.proxmox.com/debian/pve stretch pve-no-subscription`” pada file `/etc/apt/sources.list` menggunakan editor *nano*.
nano /etc/apt/sources.list

```
GNU nano 2.7.4 File: /etc/apt/sources.list Modified
deb http://ftp.debian.org/debian stretch main contrib
deb http://ftp.debian.org/debian stretch-updates main contrib
# security updates
deb http://security.debian.org stretch/updates main contrib
deb http://download.proxmox.com/debian/pve stretch pve-no-subscription
```

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan **Enter**. Tekan tombol **CTRL+X** untuk keluar dari editor *nano*.

6. Memverifikasi hasil pengaturan *PVE No-subscription repository* menggunakan perintah “`cat /etc/apt/sources.list`”.

```
root@pve:~# cat /etc/apt/sources.list
deb http://ftp.debian.org/debian stretch main contrib

deb http://ftp.debian.org/debian stretch-updates main contrib

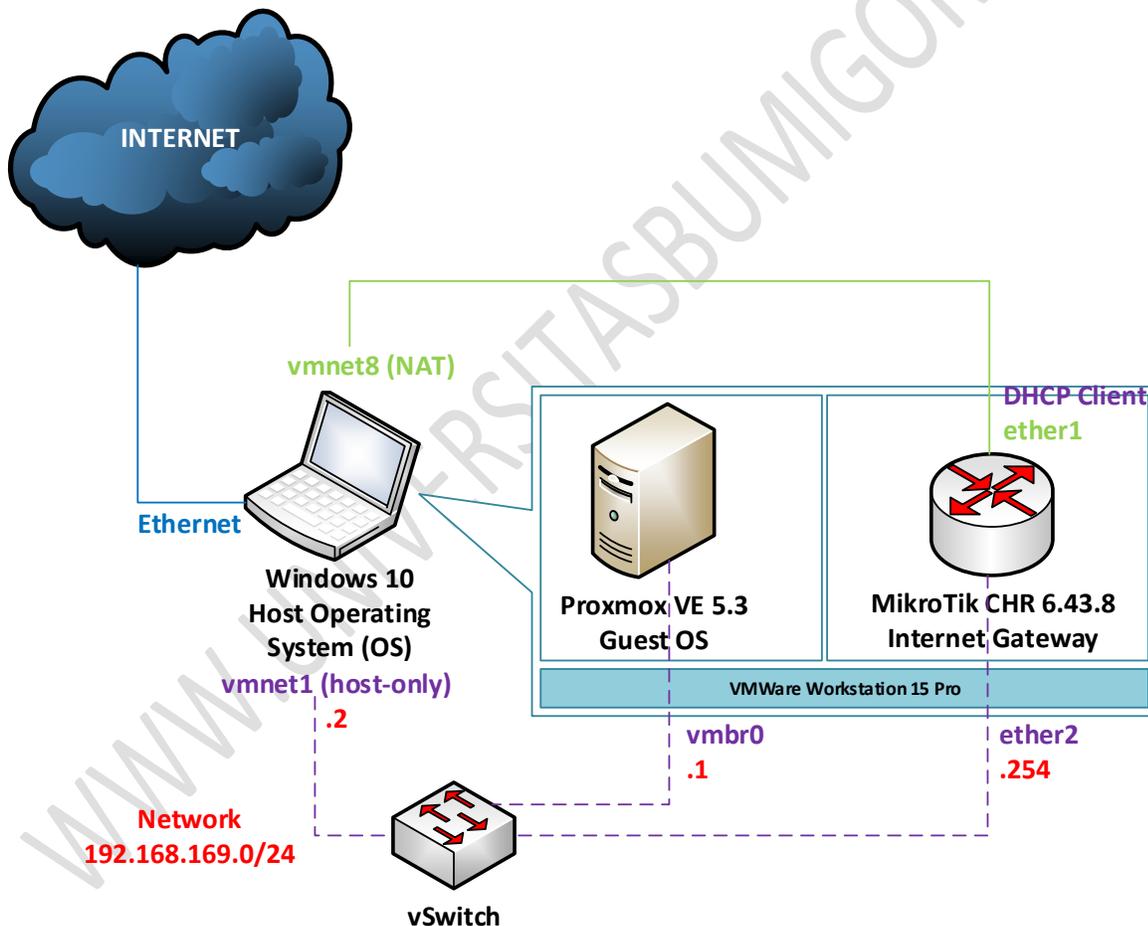
# security updates
deb http://security.debian.org stretch/updates main contrib

deb http://download.proxmox.com/debian/pve stretch pve-no-subscription
root@pve:~#
```

BAB IV

INSTALASI DAN KONFIGURASI MIKROTIK CHR SEBAGAI INTERNET GATEWAY PADA VMWARE WORKSTATION 15

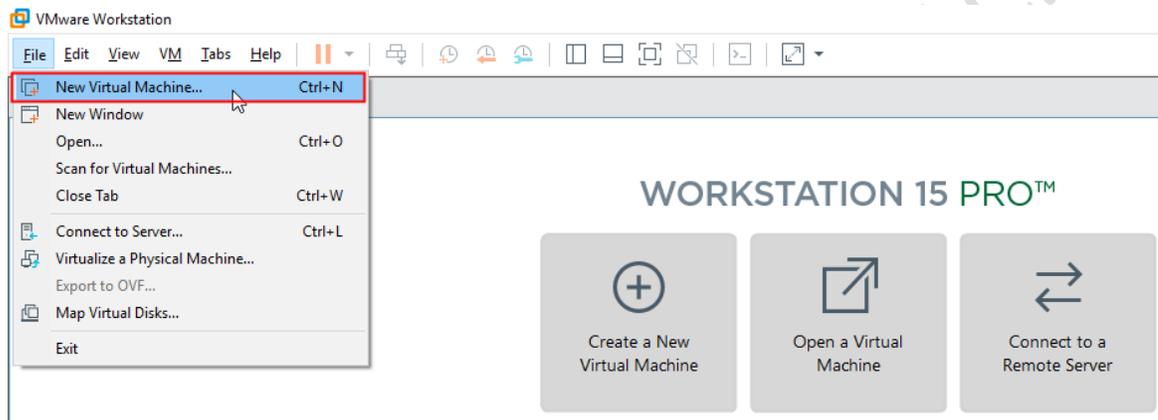
Untuk menjembatani kebutuhan akses *Internet* dari *Proxmox VE 5.3* maka pada *VMWare Workstation* akan diinstalasi **Mikrotik CHR 6.43.8** sebagai *Virtual Machine (VM)* dan dikonfigurasi sebagai gateway untuk berbagi pakai koneksi *Internet*, seperti terlihat pada gambar berikut:



MikroTik CHR memiliki 2 (dua) *interface* yaitu **ether1** dengan jenis koneksi *NAT* dan **ether2** dengan jenis koneksi *host-only*. *Interface ether2* menggunakan alamat IP **192.168.169.254/24**. Sedangkan *interface ether1* dialokasi secara dinamis sehingga bertindak sebagai **DHCP Client**.

Adapun langkah-langkah instalasi dan konfigurasi *MikroTik CHR* sebagai *gateway Internet* pada *VMWare Workstation 15 Pro* adalah sebagai berikut:

1. Jalankan aplikasi *VMWare Workstation 15 Pro* melalui **Start > VMWare > VMWare Workstation Pro**.
2. Tampil aplikasi *VMWare Workstation*. Untuk membuat *virtual machine* baru, pilih menu **File > New Virtual Machine ...**, seperti terlihat pada gambar berikut:

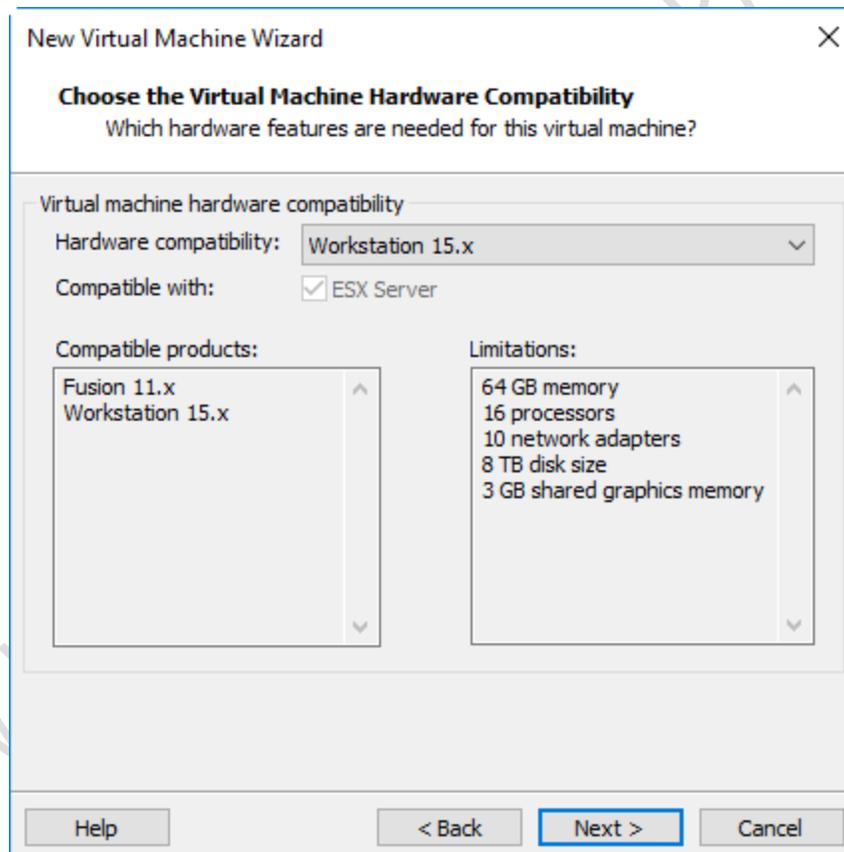


3. Tampil kotak dialog *New Virtual Machine Wizard* untuk menentukan jenis konfigurasi *virtual machine* yang ingin dibuat, seperti terlihat pada gambar berikut:



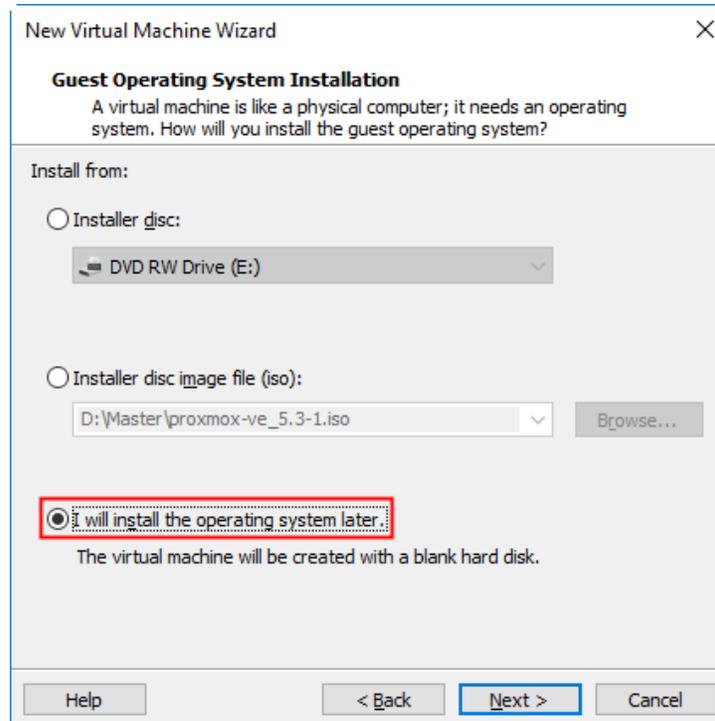
Terdapat 2 pilihan jenis konfigurasi yang dapat dipilih yaitu *Typical (recommended)* dan *Custom (advanced)*. Jenis konfigurasi *Typical* disarankan untuk dipilih ketika ingin membuat virtual machine melalui beberapa tahapan dengan mudah. Sebaliknya jenis konfigurasi *Custom* akan memberikan pilihan pengaturan lanjutan seperti penentuan jenis *controller SCSI*, jenis *virtual disk* dan kompatibilitas dengan produk *VMWare* versi sebelumnya. Pilih **Custom (advanced)**, dan klik tombol **Next >** untuk melanjutkan.

4. Tampil kotak dialog *Choose the Virtual Machine Hardware Compatibility* untuk menentukan kompatibilitas perangkat keras dari *virtual machine*, seperti terlihat pada gambar berikut:



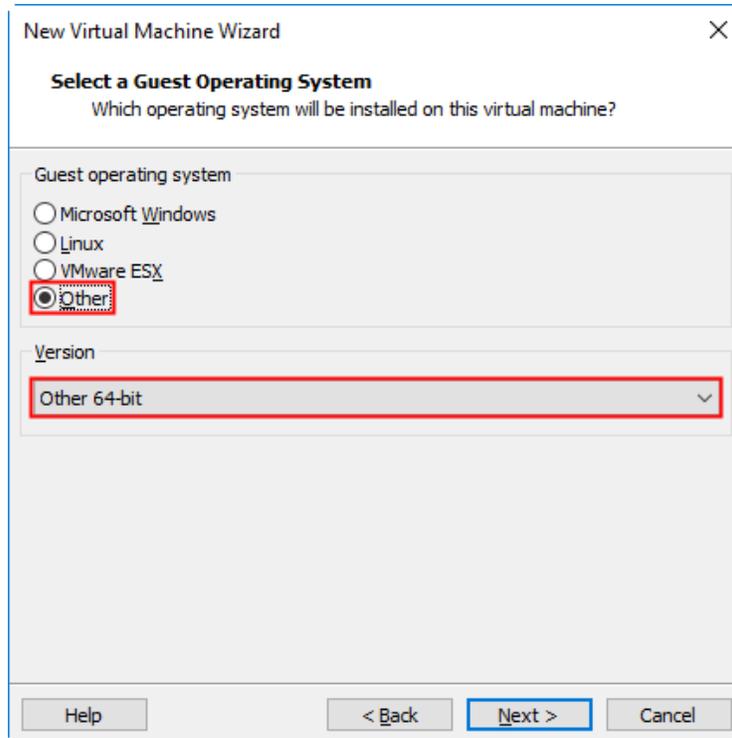
Secara *default* telah terpilih **Workstation 15.x** pada pilihan parameter *Hardware compatibility*. Klik tombol **Next >** untuk melanjutkan.

5. Tampil kotak dialog *Guest Operating System Installation* untuk menentukan bagaimana cara instalasi sistem operasi dilakukan, seperti terlihat pada gambar berikut:



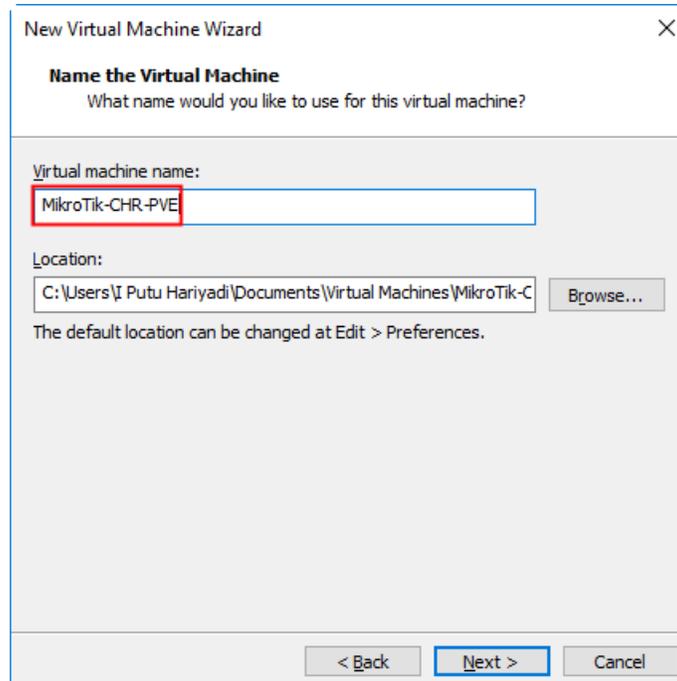
Terdapat 3 pilihan yaitu *Install from Installer disc* untuk menginstalasi dari media disc seperti CD/DVD, *Install from Installer disc image file (iso)* untuk menginstalasi dari file ISO, dan *I will install the operating system later* untuk mempersiapkan virtual machine dengan hardisk kosong tanpa melakukan instalasi sistem operasi. Pilih *I will install the operating system later*, dan klik tombol **Next >** untuk melanjutkan.

6. Tampil kotak dialog *Select a Guest Operating System* untuk menentukan jenis sistem operasi yang akan diinstalasi pada virtual machine yang dibuat, seperti ditunjukkan pada gambar berikut:



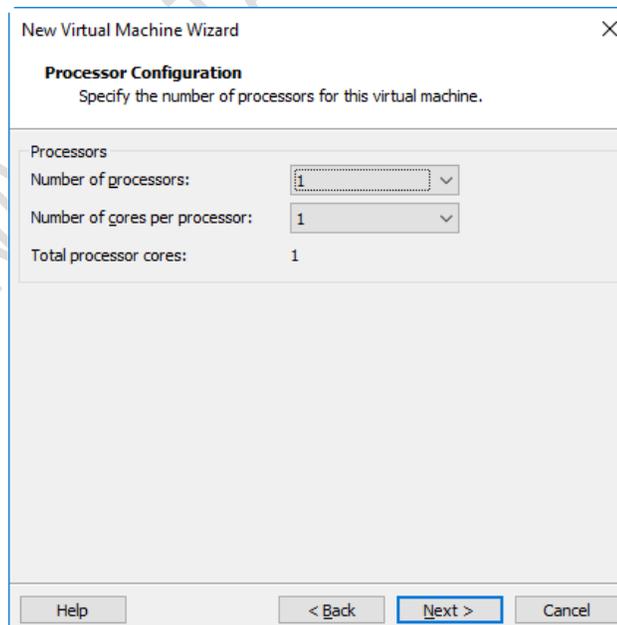
Pilih *Other* pada bagian *Guest operating system* dan *Other 64-bit* pada bagian *Version*. Klik tombol **Next >** untuk melanjutkan.

7. Tampil kotak dialog *Name the Virtual Machine* untuk menentukan nama pengenal *virtual machine* dan menentukan lokasi penyimpanan file *virtual machine* yang dibuat. Pada bagian *Virtual machine name* masukkan nama pengenal virtual machine, sebagai contoh **MikroTik-CHR-PVE**. Apabila diperlukan pada bagian *Location* dapat pula ditentukan lokasi penyimpanan file virtual machine yang dibuat dengan cara menekan tombol *Browse* ..., seperti terlihat pada gambar berikut:



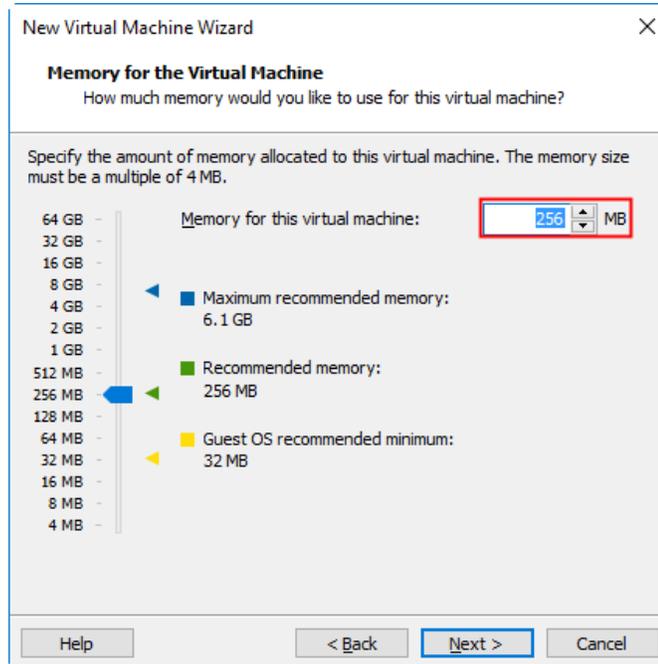
Klik tombol **Next >** untuk melanjutkan.

8. Tampil kotak dialog *Processor Configuration* untuk menentukan jumlah prosesor yang digunakan oleh virtual machine yang dibuat, seperti terlihat pada gambar berikut:



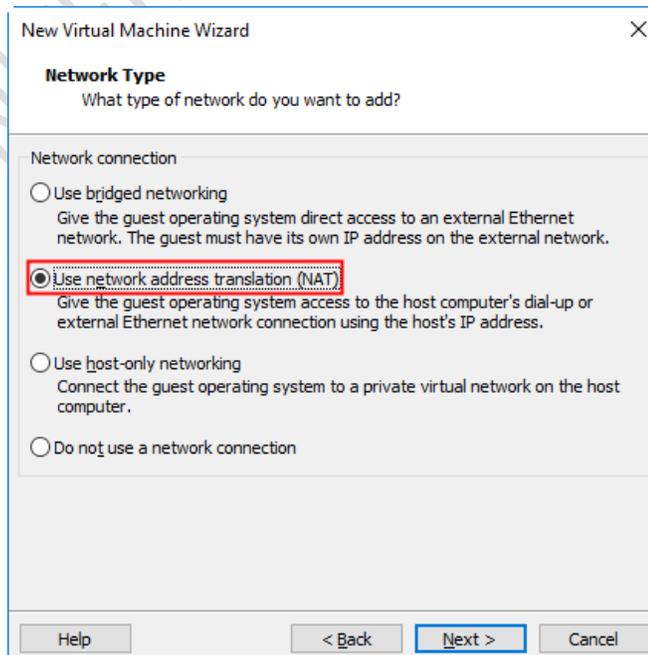
Secara *default* telah terpilih **1 (satu)** baik untuk jumlah prosesor maupun jumlah *core* per prosesor. Klik tombol **Next >** untuk melanjutkan.

9. Tampil kotak dialog *Memory for the Virtual Machine* untuk menentukan kapasitas memori yang dialokasikan untuk *virtual machine* yang dibuat. Pada isian dari parameter *Memory for this virtual machine*, masukkan **256 MB**, seperti terlihat pada gambar berikut:



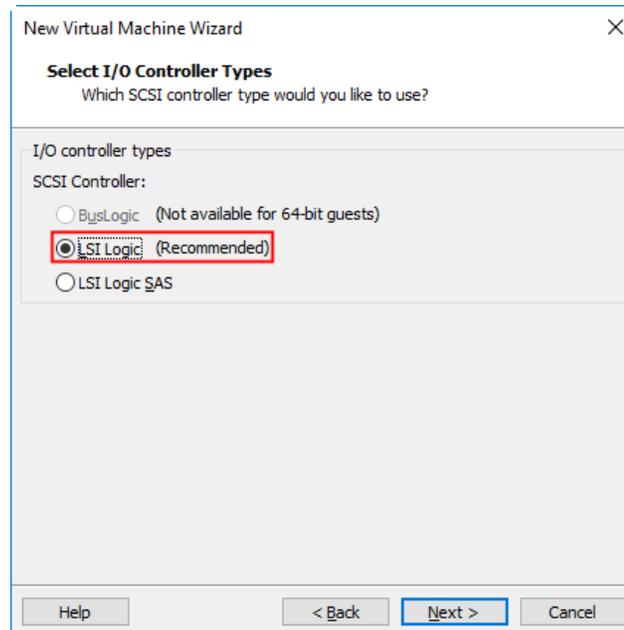
Klik tombol **Next >** untuk melanjutkan.

10. Tampil kotak dialog *Network Type* untuk menentukan jenis koneksi jaringan yang digunakan, seperti terlihat pada gambar berikut:



Pilih *use network address translation (NAT)*. Klik tombol **Next >** untuk melanjutkan.

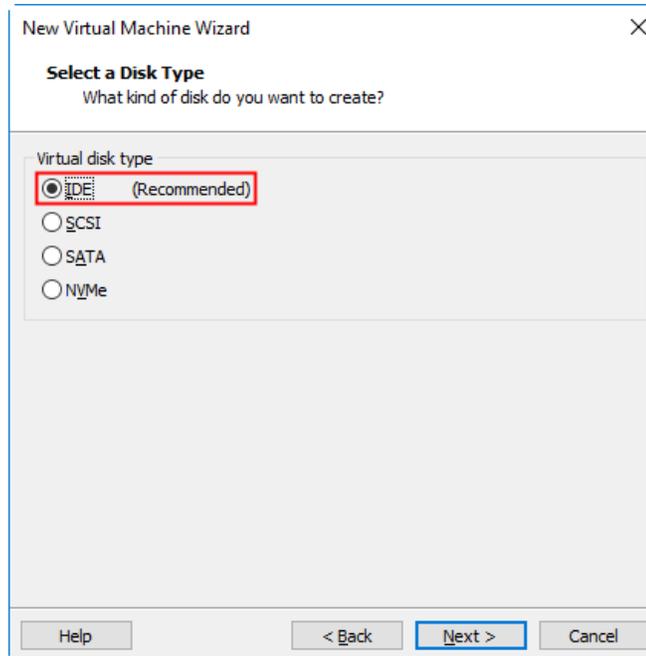
11. Tampil kotak dialog *Select I/O Controller Types* untuk menentukan jenis SCSI controller yang ingin digunakan, seperti terlihat pada gambar berikut:



Secara *default* telah terpilih *LSI Logic* yang merupakan pilihan yang direkomendasikan.

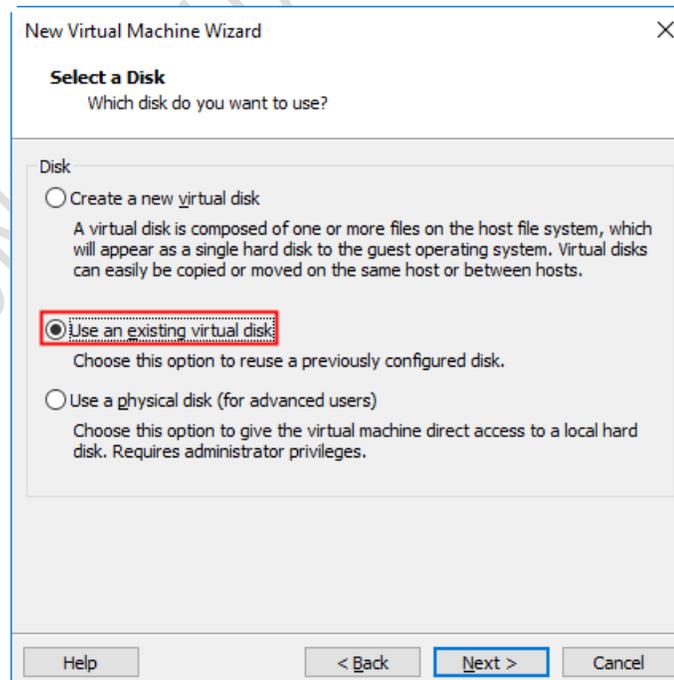
Klik tombol **Next >** untuk melanjutkan.

12. Tampil kotak dialog *Select a Disk Type* untuk menentukan jenis disk yang akan dibuat, seperti terlihat pada gambar berikut:



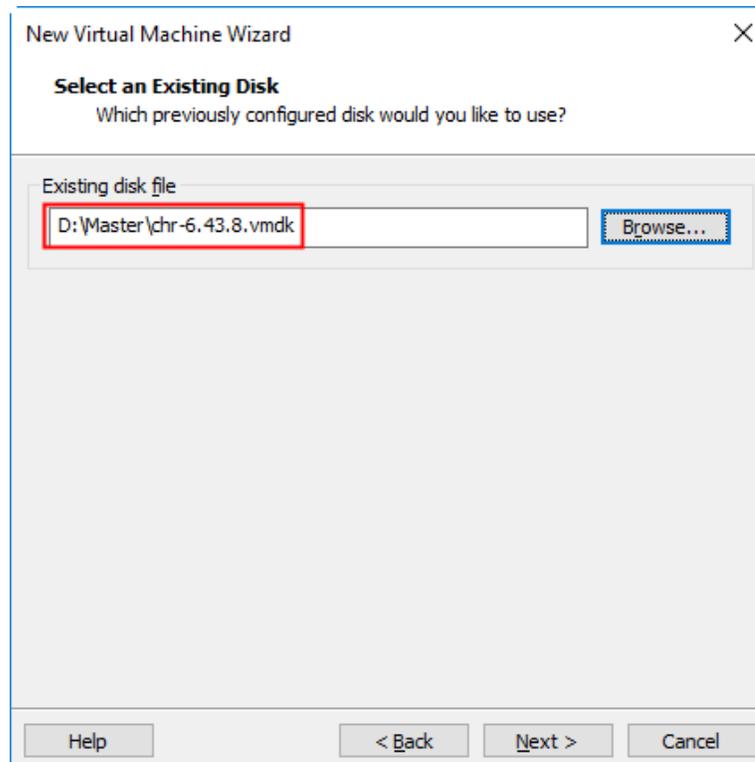
Secara default telah terpilih **IDE** yang merupakan pilihan yang direkomendasikan. Klik tombol **Next >** untuk melanjutkan.

13. Tampil kotak dialog *Select a Disk* untuk menentukan *disk* yang ingin digunakan, seperti terlihat pada gambar berikut:



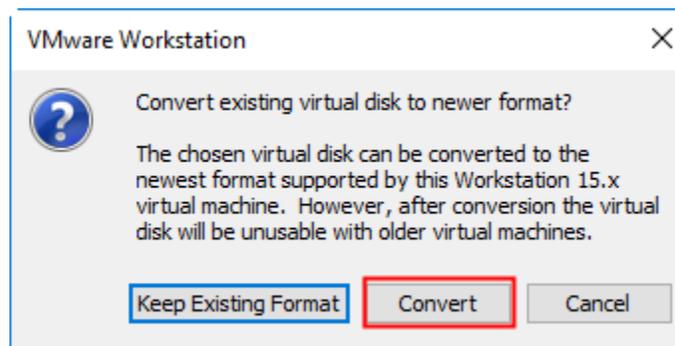
Pilih *Use an existing virtual disk* untuk menggunakan *virtual disk* yang telah ada yaitu **chr-6.43.8.vmdk** yang telah diunduh dari situs [Mikrotik](#). Klik tombol **Next >** untuk melanjutkan.

14. Tampil kotak dialog *Select an existing disk* untuk mengarahkan ke lokasi virtual disk yang ingin digunakan. Klik tombol *Browse...* dan arahkan ke lokasi penyimpanan file **chr-6.43.8.vmdk**, sebagai contoh berada di **D:\Master**, seperti terlihat pada gambar berikut:



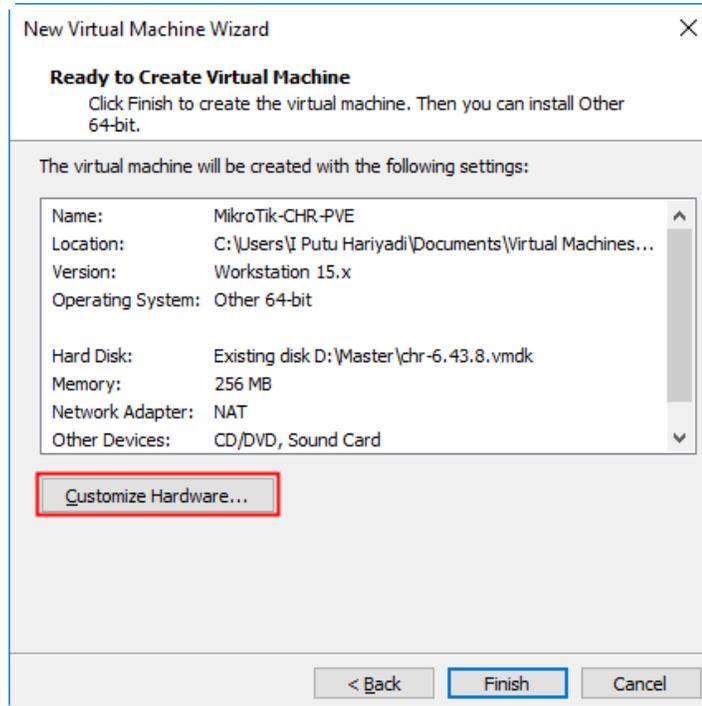
Klik tombol **Next >** untuk melanjutkan.

15. Tampil kotak dialog *Convert existing virtual disk to newer format?* yang menginformasikan bahwa virtual disk terpilih dapat dikonversi ke format lebih baru yang di dukung oleh *Workstation 15.x*, seperti terlihat pada gambar berikut:

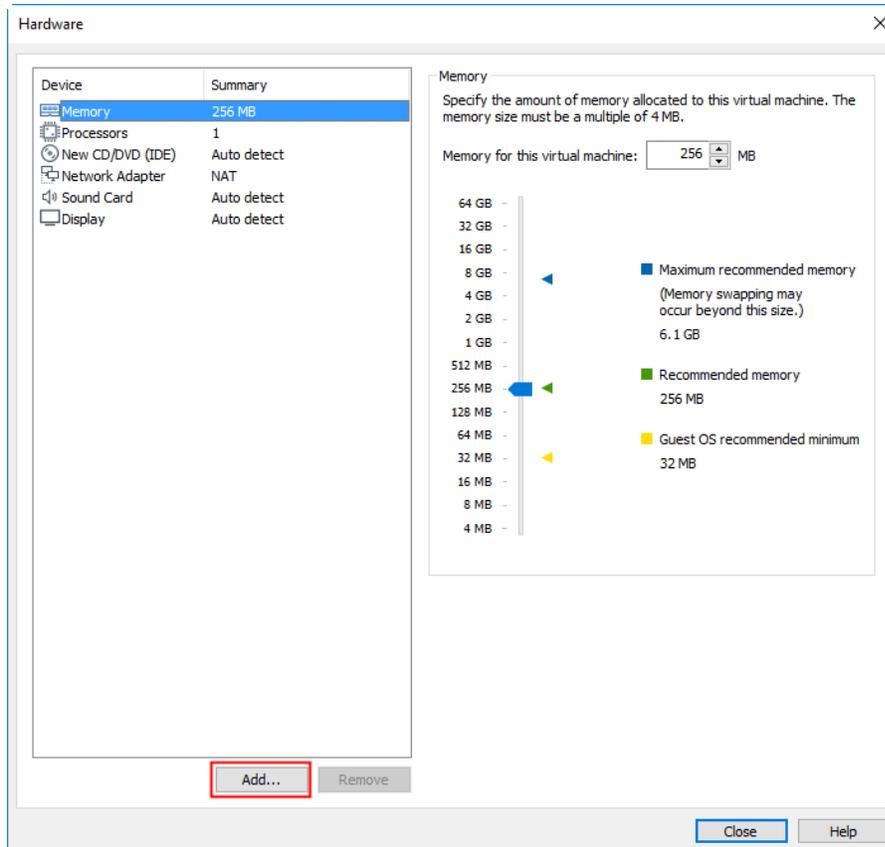


Klik tombol **Convert** untuk mengkonversi ke format *Workstation 15.x*.

16. Tampil kotak dialog *Ready to Create Virtual Machine* yang memperlihatkan ringkasan pengaturan *virtual machine* yang akan dibuat, seperti terlihat pada gambar berikut:

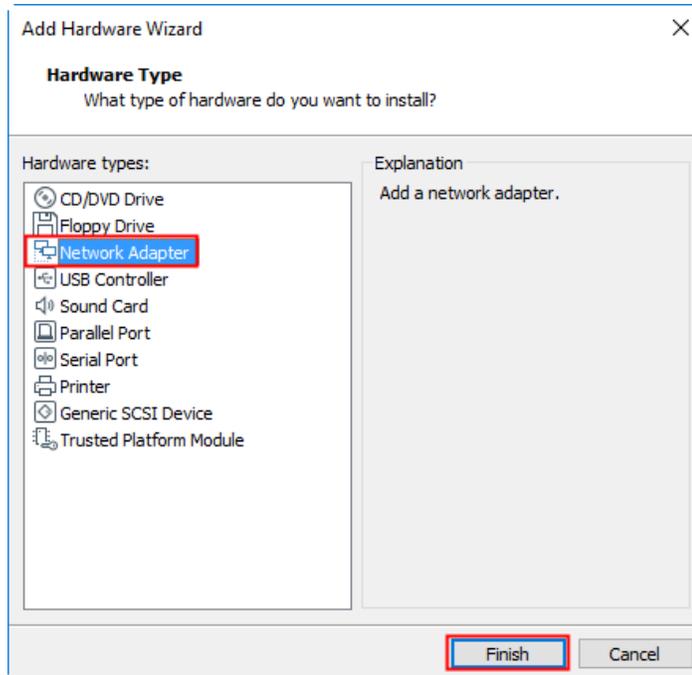


Klik tombol **Customize Hardware...** untuk menambahkan *network adapter* kedua maka akan tampil kotak dialog *Hardware*, seperti terlihat pada gambar berikut:

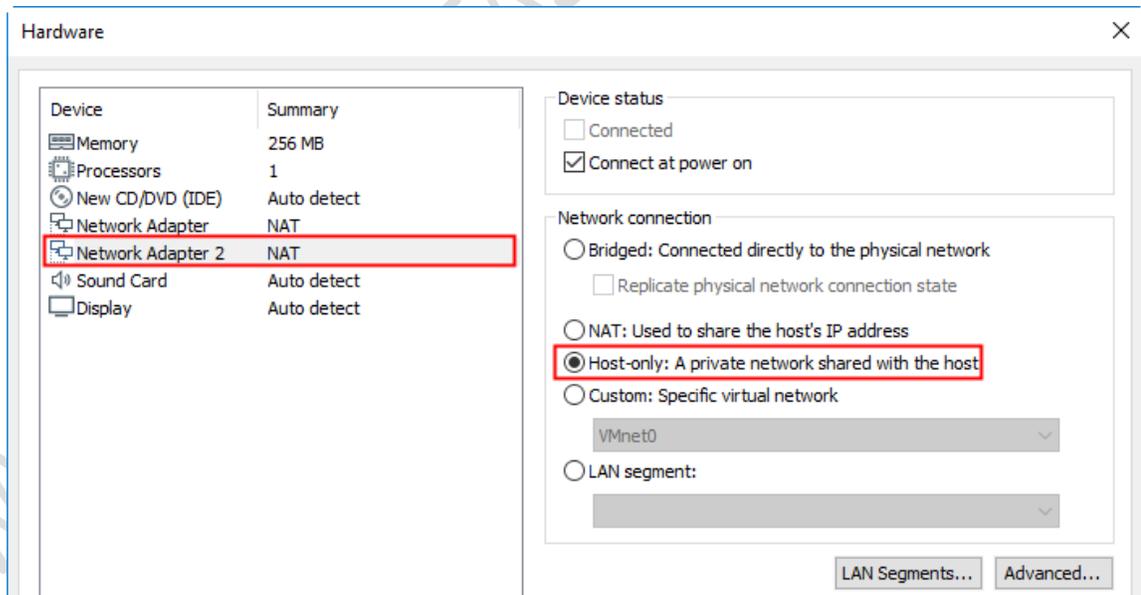


Klik tombol **Add**.

Tampil kotak dialog *Hardware Type* yang digunakan untuk menentukan jenis perangkat keras yang ingin diinstalasi, seperti terlihat pada gambar berikut:



Pilih **Network Adapter** pada pilihan *Hardware types*: dan klik tombol **Finish** maka akan tampil kembali kotak dialog *Hardware* yang telah memperlihatkan hasil penambahan **Network Adapter 2**, seperti terlihat pada gambar berikut:

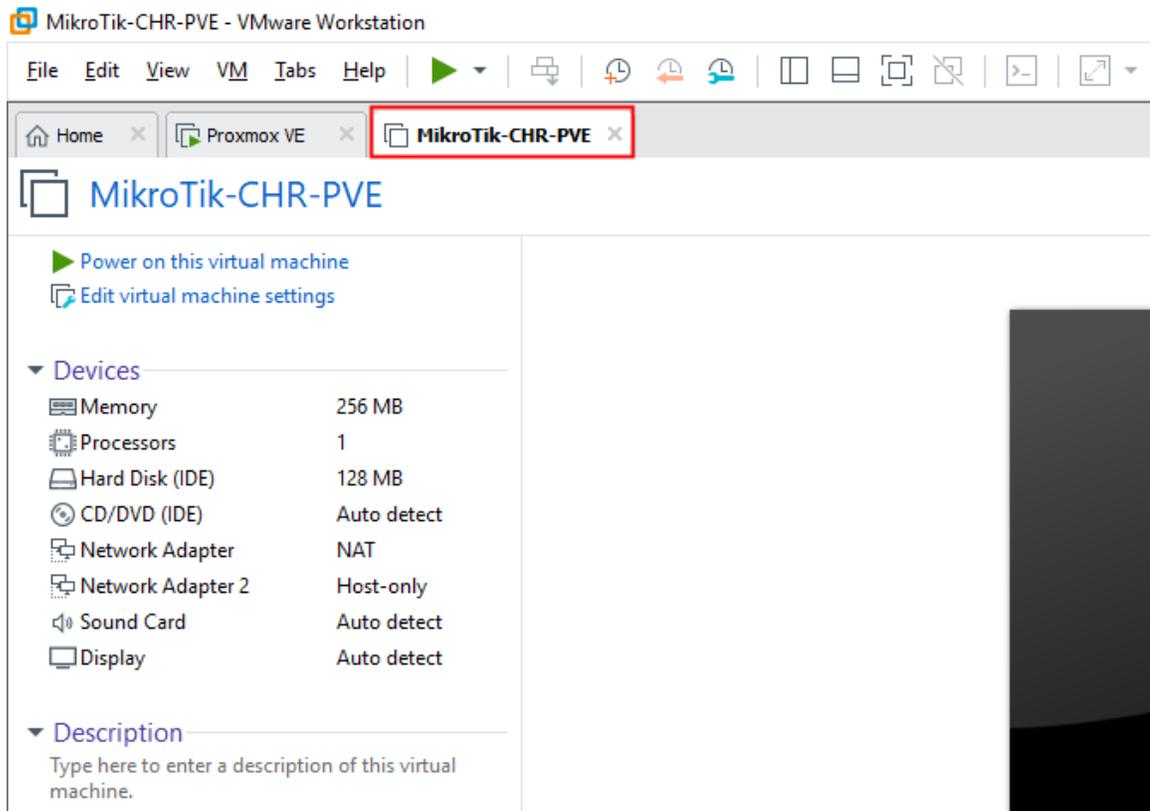


Lakukan perubahan network connection dari Network Adapter 2 menjadi Host-only.

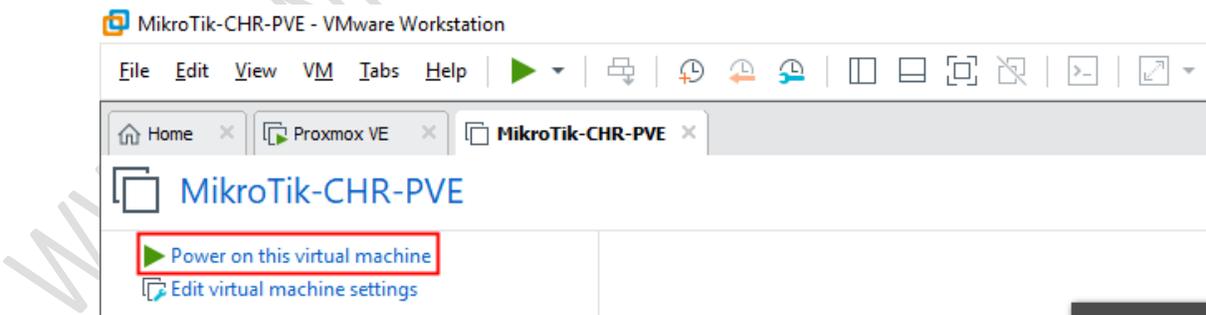
Klik tombol **Close** untuk menutup kotak dialog *Hardware*.

Klik tombol **Finish** untuk membuat virtual machine.

17. Tampil kotak dialog yang menampilkan *virtual machine* yang telah berhasil dibuat yaitu dengan nama pengenal **MikroTik-CHR-PVE**, seperti terlihat pada gambar berikut:



- Selanjutnya klik **Power on the virtual machine** untuk menghidupkan *virtual machine* seperti terlihat pada gambar berikut:



18. Tampil inputan **Mikrotik Login** untuk proses otentikasi sebelum pengguna dapat mengakses **Command Line Interface (CLI)** dari *Mikrotik*, seperti terlihat pada gambar berikut:

```
MikroTik 6.43.8 (stable)
MikroTik Login: _
```

Masukkan nama login “**admin**” pada inputan **MikroTik Login** dan tekan tombol **Enter**.

Selanjutnya tampil inputan **Password:**. Tekan tombol **Enter** untuk melanjutkan karena *password* untuk user “*admin*” adalah **kosong (blank)**, seperti terlihat pada gambar berikut:

```
MikroTik 6.43.8 (stable)
MikroTik Login: admin
Password:
```

Selanjutnya tampil pesan “**Do you want to see the software license? [Y/n]**”, tekan “**n**” untuk tidak menampilkan lisensi perangkat lunak. Terlihat *prompt CLI* dari *Mikrotik*, seperti gambar berikut:

```
[admin@MikroTik] >
```

19. Menampilkan informasi *interface* yang dimiliki oleh router Mikrotik.

```
[admin@MikroTik] > interface print
Flags: D - dynamic, X - disabled, R - running, S - slave
#   NAME      TYPE      ACTUAL-MTU  LZMTU
0   R ether1    ether      1500
1   R ether2    ether      1500
```

Terlihat terdapat 2 (dua) *interface* yaitu **ether1** dan **ether2**.

20. Menampilkan informasi interface dengan pengaturan *DHCP Client*.

```
[admin@MikroTik] > ip dhcp-client print
Flags: X - disabled, I - invalid, D - dynamic
#   INTERFACE      USE ADD-DEFAULT-ROUTE  STATUS      ADDRESS
0   ether2          yes yes                 bound       192.168.195.145/24
```

Terlihat terdapat satu interface yaitu **ether2**.

21. Menghapus *interface ether2* dari pengaturan *DHCP Client* dan memverifikasi hasil penghapusannya.

```
[admin@MikroTik] > ip dhcp-client remove 0
[admin@MikroTik] > ip dhcp-client print
Flags: X - disabled, I - invalid, D - dynamic
#   INTERFACE      USE ADD-DEFAULT-ROUTE  STATUS      ADDRESS
```

22. Mengatur *DHCP Client* pada *interface ether1*.

```
[admin@MikroTik] > ip dhcp-client add interface=ether1 disabled=no
```

23. Memverifikasi hasil pengaturan *DHCP Client* pada *interface ether1*.

```
[admin@MikroTik] > ip dhcp-client print
Flags: X - disabled, I - invalid, D - dynamic
#   INTERFACE          USE ADD-DEFAULT-ROUTE STATUS      ADDRESS
0   ether1              yes yes          bound      192.168.163.135/24
```

24. Mengatur pengalamatan IP pada interface ether2 secara static menggunakan 192.168.169.254/24.

```
[admin@MikroTik] > ip address add address=192.168.169.254/24 interface=ether2
```

25. Memverifikasi pengalamatan IP yang telah diatur pada interface.

```
[admin@MikroTik] > ip address print
Flags: X - disabled, I - invalid, D - dynamic
#   ADDRESS          NETWORK      INTERFACE
0   D 192.168.163.135/24 192.168.163.0 ether1
1   192.168.169.254/24 192.168.169.0 ether2
```

26. Menampilkan informasi tabel routing.

```
[admin@MikroTik] > ip route print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
#   DST-ADDRESS      PREF-SRC      GATEWAY      DISTANCE
0   ADS 0.0.0.0/0        192.168.163.2 1
1   ADC 192.168.163.0/24 192.168.163.135 ether1        0
2   ADC 192.168.169.0/24 192.168.169.254 ether2        0
```

27. Memverifikasi koneksi ke default gateway menggunakan perintah ping.

```
[admin@MikroTik] > ping 192.168.163.2
SEQ HOST          SIZE TTL TIME STATUS
0 192.168.163.2   56 128 0ms
1 192.168.163.2   56 128 0ms
2 192.168.163.2   56 128 0ms
3 192.168.163.2   56 128 0ms
sent=4 received=4 packet-loss=0% min-rtt=0ms avg-rtt=0ms max-rtt=0ms
```

Tekan **CTRL+C** untuk menghentikan ping.

28. Mengatur agar Mikrotik bertindak sebagai Server DNS.

```
[admin@MikroTik] > ip dns set allow-remote-requests=yes
```

29. Memverifikasi pengaturan Server DNS.

```
[admin@MikroTik] > ip dns print
servers:
dynamic-servers: 192.168.163.2
allow-remote-requests: yes
max-udp-packet-size: 4096
query-server-timeout: 2s
query-total-timeout: 10s
max-concurrent-queries: 100
max-concurrent-tcp-sessions: 20
cache-size: 2048KiB
cache-max-ttl: 1w
cache-used: 17KiB
```

30. Mengatur *Internet Connection Sharing (ICS)* menggunakan *Network Address Translation (NAT)*.

```
[admin@MikroTik] > ip firewall nat add chain=srcnat out-interface=ether1 action=masquerade
```

31. Memverifikasi hasil pengaturan NAT.

```
[admin@MikroTik] > ip firewall nat print
Flags: X - disabled, I - invalid, D - dynamic
0 chain=srcnat action=masquerade out-interface=ether1
```

32. Memverifikasi koneksi Internet menggunakan perintah ping ke salah satu situs di Internet.

```
[admin@MikroTik] > ping detik.com
  SEQ HOST                                SIZE TTL TIME  STATUS
  0 203.190.242.211                        56 128 46ms
  1 203.190.242.211                        56 128 78ms
  2 203.190.242.211                        56 128 47ms
sent=3 received=3 packet-loss=0% min-rtt=46ms avg-rtt=57ms max-rtt=78ms
```

Tekan **CTRL+C** untuk menghentikan *ping*.

BAB V

INSTALASI DAN KONFIGURASI LINUX CONTAINER (LXC)

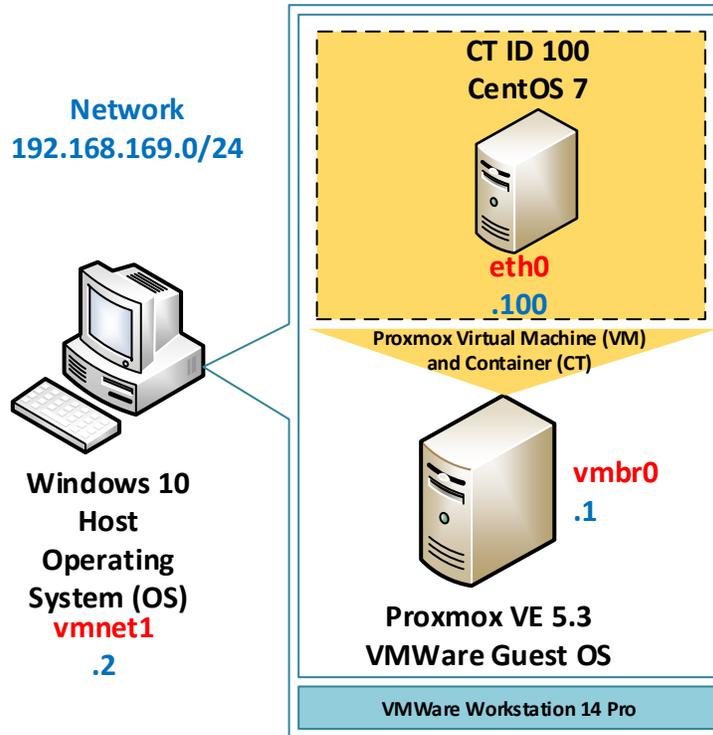
CENTOS 7 PADA PROXMOX VE 5.3

Pada bab ini akan dibahas penerapan teknologi virtualisasi yang didukung oleh *Proxmox VE* yaitu **Linux Container (LXC)** menggunakan **CentOS 7**. Menurut situs [Proxmox](#), **LXC** merupakan lingkungan virtualisasi level sistem operasi untuk menjalankan beberapa sistem Linux terisolasi pada sebuah kontrol host Linux. *LXC* menjadi alternatif dari *full machine virtualization* yang menawarkan *low overhead*. *Container* akan menggunakan sistem operasi dari *host* daripada mengemulasikan sistem operasi secara lengkap sehingga berdampak pada keseluruhan *container* menggunakan *kernel* yang sama dan dapat mengakses sumber daya secara langsung dari *host*. Pengguna Linux dapat membuat dan manajemen *container* sistem atau aplikasi menggunakan *Application Programming Interface (API)*.

Pembahasan pada bab ini terdiri dari 2 (dua) bagian yaitu (a) Rancangan Jaringan Ujicoba, (b) Instalasi dan Konfigurasi *LXC CentOS 7* pada *Proxmox VE*.

A. RANCANGAN JARINGAN UJICOBA

Rancangan jaringan ujicoba yang digunakan, seperti terlihat pada gambar berikut:

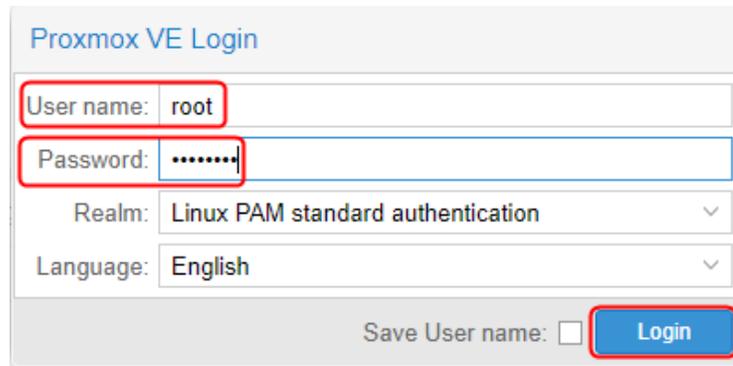


Pada *Server Proxmox VE 5.3* akan dilakukan pembuatan *Container (CT)* dengan sistem operasi *CentOS 7* dan menggunakan ID **100** serta alamat IP **192.168.169.100/24**.

B. INSTALASI DAN KONFIGURASI LXC CENTOS 7 PADA PROXMOX VE

Adapun langkah-langkah instalasi dan konfigurasi *LXC CentOS 7* pada *Proxmox VE* adalah sebagai berikut:

1. Buka *browser*, sebagai contoh menggunakan **Chrome**. Pada *address bar* dari browser, masukkan URL <https://192.168.169.1:8006>.
2. Tampil kotak dialog otentikasi *Proxmox VE Login*, lengkapi isian "**User name**" dan "**Password**". Pada isian "*User name*", masukkan "**root**". Sedangkan pada isian "*Password*", masukkan sandi login dari user "*root*" yaitu **12345678**, seperti terlihat pada gambar berikut:



Proxmox VE Login

User name: root

Password:

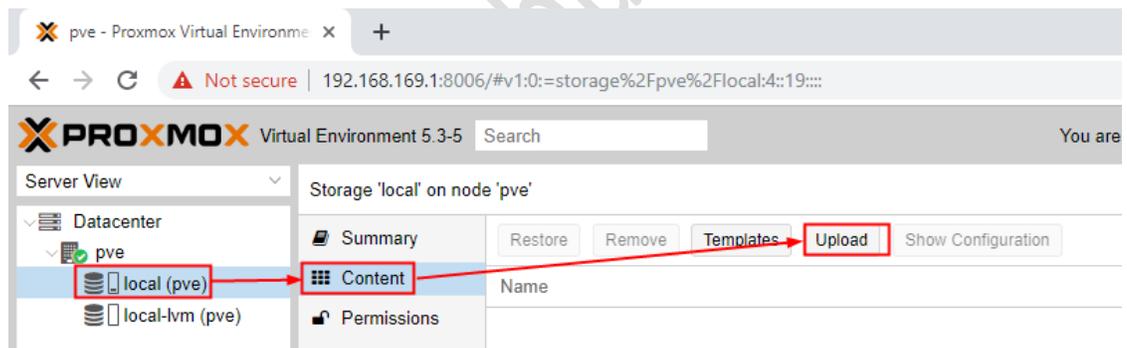
Realm: Linux PAM standard authentication

Language: English

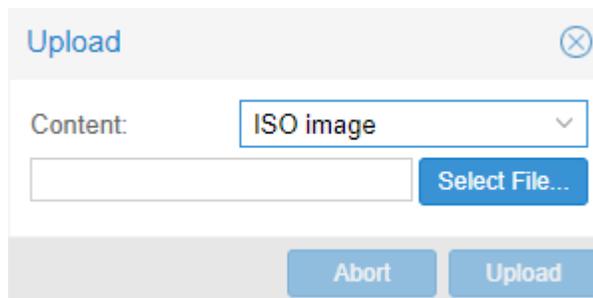
Save User name: Login

Klik tombol **Login**. Pengguna langsung diarahkan ke tampilan halaman *Server View* dari *Proxmox*.

- Mengunggah file **template container** ke *Server Proxmox* dengan mengakses node "**pve**" di bawah menu **Datacenter** pada panel sebelah kiri dan memilih *storage local (pve)*. Pilih menu **Content** pada panel sebelah kanan dari *local (pve)* dan pilih **Upload** untuk mengunggah *file template container CentOS 7*, seperti terlihat pada gambar berikut:



Tampil kotak dialog **Upload**, seperti terlihat pada gambar berikut:



Upload

Content: ISO image

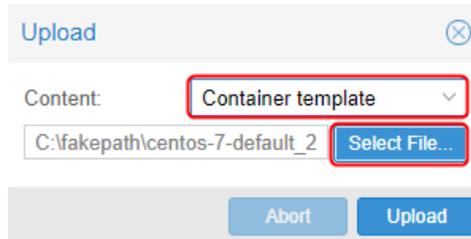
Select File...

Abort Upload

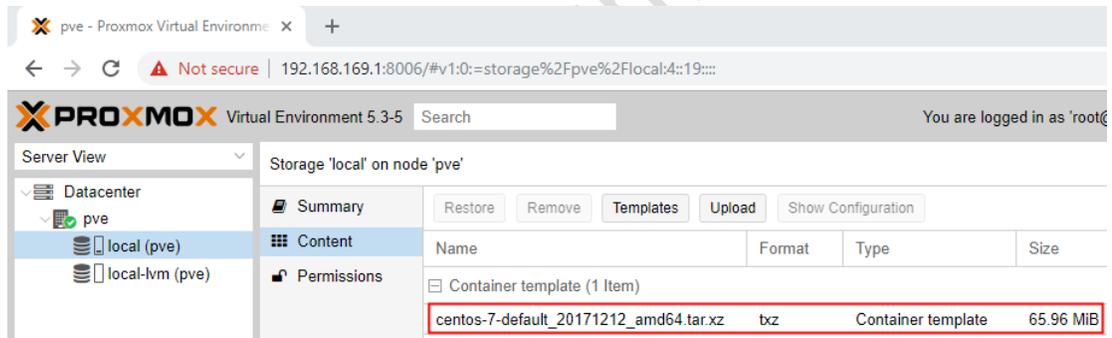
Terdapat beberapa parameter yang harus diatur yaitu:

- Content**:, pilih **Container Template**.

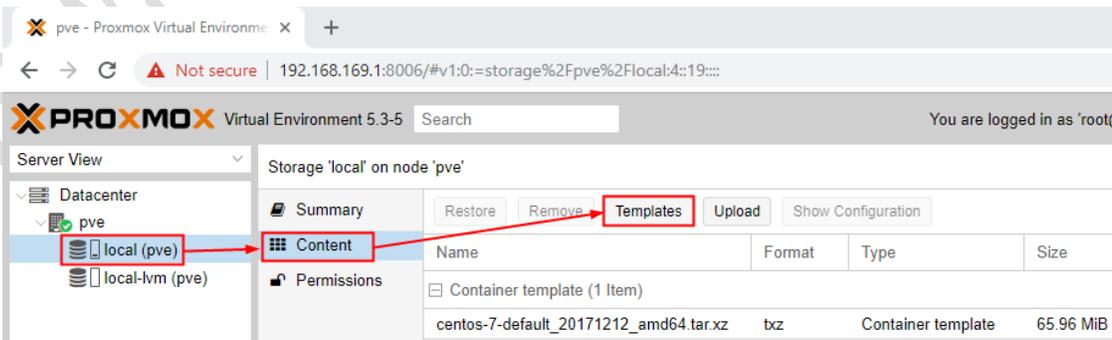
- b) Tekan tombol **Select File...** untuk mengarahkan ke lokasi direktori penyimpanan file **template CentOS 7**, sebagai contoh di **D:\Master\centos-7-default_20170504_amd64.tar.xz**, seperti terlihat pada gambar berikut:



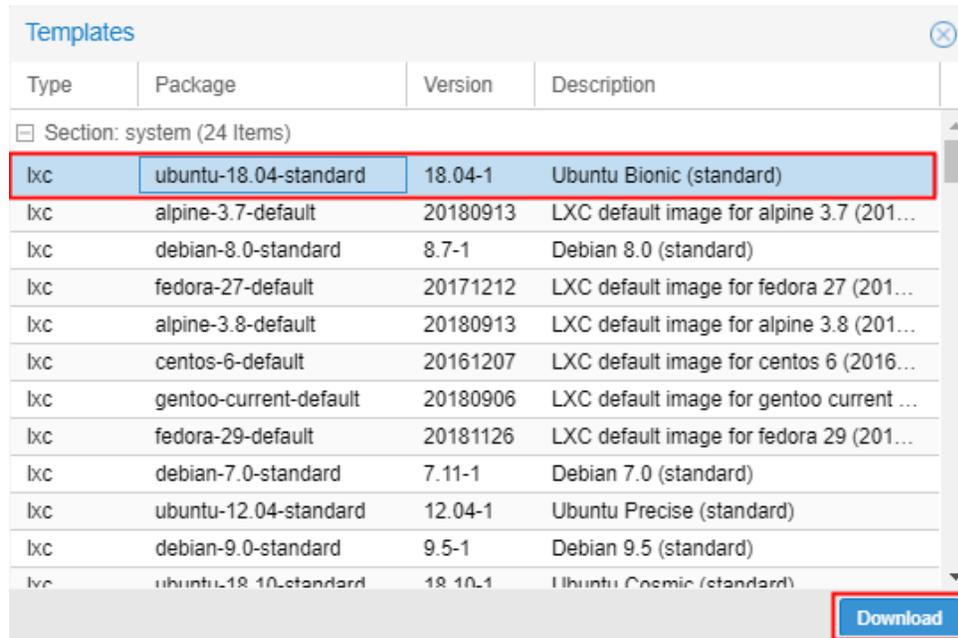
Tekan tombol **Upload** dan tunggu hingga proses pengunggahan file selesai dilakukan. Apabila proses unggah berhasil dilakukan maka pada bagian **Content** dari **storage local (pve)** akan menampilkan nama file **centos-7-default_20170504_amd64.tar.xz**, seperti terlihat pada gambar berikut:



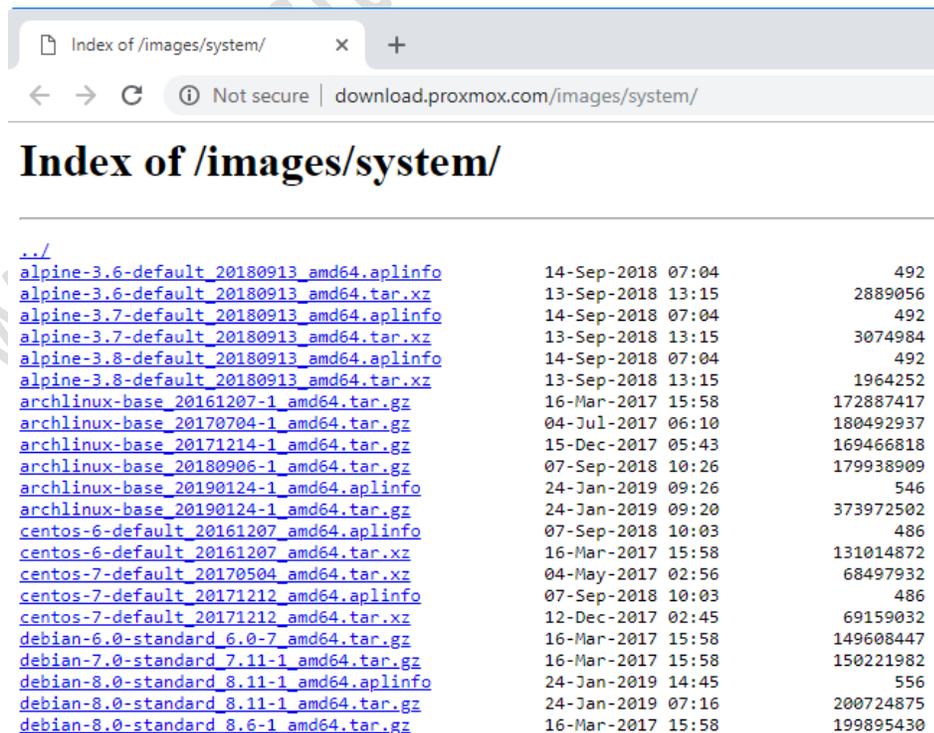
File *container template* juga dapat diunduh langsung dari *Internet* dengan menekan tombol **Templates** di bagian **Content** dari **storage local (pve)**, seperti terlihat pada gambar berikut:



Pada kotak dialog **Templates** yang tampil, pilih **package** yang ingin diunduh dan tekan tombol **Download**, seperti terlihat pada gambar berikut:

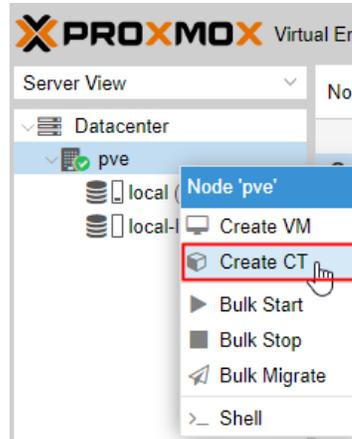


Tunggu hingga proses unduh selesai dilakukan. Atau file *template container* juga dapat diunduh secara manual melalui alamat <http://download.proxmox.com/images/system/>, seperti terlihat pada gambar berikut:



Selanjutnya file *template container* yang telah diunduh dapat diunggah ke *Server Proxmox* dengan mengikuti langkah-langkah proses unggah *template CentOS7* yang telah dijelaskan sebelumnya.

4. Membuat **Container** dengan cara klik kanan pada *node "pve"* dibawah menu **Datacenter** di panel sebelah kiri dan memilih **Create CT**, seperti terlihat pada gambar berikut:



Tampil kotak dialog **Create: LXC Container**. Terdapat beberapa parameter yang diatur di bagian **General** dari **LXC Container**, seperti terlihat pada gambar berikut:

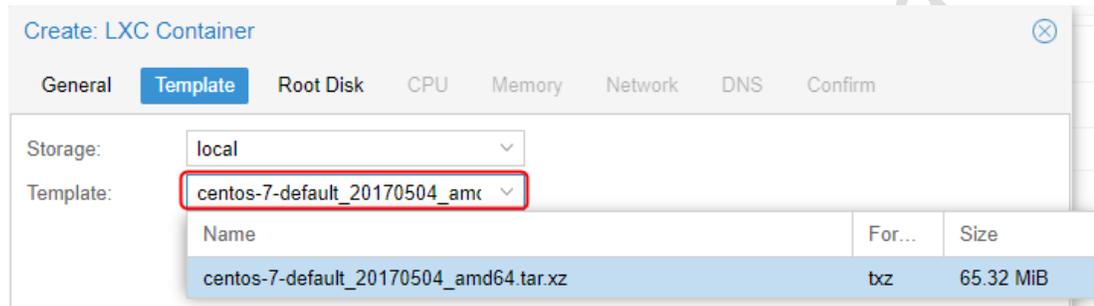
 A screenshot of the 'Create: LXC Container' dialog box in Proxmox VE. The 'General' tab is selected. The following fields are visible:

- Node:** pve (dropdown menu)
- CT ID:** 100 (text input, highlighted with a red box)
- Hostname:** server100.ubg.local (text input, highlighted with a red box)
- Unprivileged container:**
- Resource Pool:** (empty dropdown menu)
- Password:** (password field, highlighted with a red box)
- Confirm password:** (password field, highlighted with a red box)
- SSH public key:** (empty text area)
- Load SSH Key File:** (button)

 At the bottom, there are 'Help', 'Advanced' (checkbox), 'Back', and 'Next' buttons.

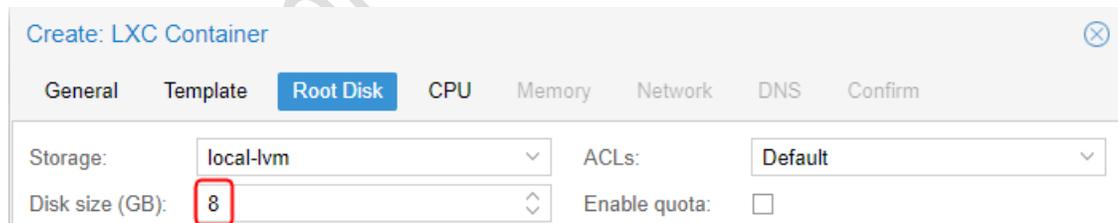
Pada parameter **Hostname:**, masukkan nama komputer dan nama domain dari *Container CentOS 7*, sebagai contoh **“server100.ubg.local”**. Sedangkan pada parameter **Password:** dan **Confirm password:**, masukkan sandi login dari user **“root”** untuk *container CentOS 7*, sebagai contoh **“12345678”**. Klik tombol **Next** untuk melanjutkan.

Tampil kotak dialog pengaturan bagian **Template** dari **LNX Container**. Pilih **centos-7-default_20170504_amd64.tar.xz** pada parameter **Template:**, seperti terlihat pada gambar berikut:



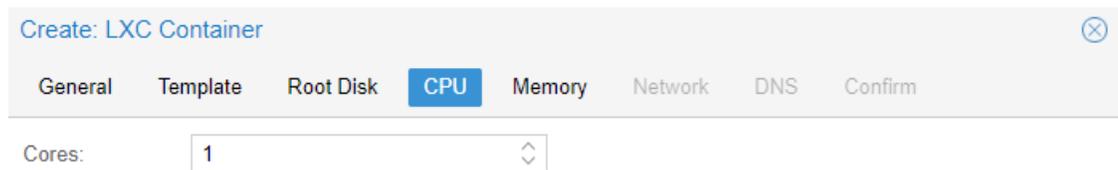
Klik tombol **Next** untuk melanjutkan.

Tampil kotak dialog pengaturan bagian **Root Disk** dari **LNX Container**. Lakukan penyesuaian ukuran hardisk yang digunakan pada parameter **Disk size (GB):**, sebagai contoh menggunakan 8 GB, seperti terlihat pada gambar berikut:



Klik tombol **Next** untuk melanjutkan.

Tampil kotak dialog pengaturan **CPU** dari **LNX Container**. Pada parameter **Cores:**, lakukan penyesuaian jumlah *Core CPU* yang digunakan apabila diperlukan. Secara *default* bernilai 1 (satu), seperti terlihat pada gambar berikut:



Klik tombol **Next** untuk melanjutkan.

Tampil kotak dialog pengaturan **Memory** dari **LNX Container**. Terdapat 2 (dua) parameter yang dapat diatur yaitu **Memory (MB)** dan **Swap (MB)**. Secara default masing-masing parameter tersebut bernilai 512 MB. Sebagai contoh akan dilakukan penyesuaian hanya pada kapasitas memori dari **512 MB** menjadi **1024 MB**, seperti terlihat pada gambar berikut:

The screenshot shows the 'Create: LXC Container' dialog box with the 'Memory' tab selected. The 'Memory (MB)' field is highlighted with a red box and contains the value '1024'. The 'Swap (MB)' field contains the value '512'.

Klik tombol **Next** untuk melanjutkan.

Tampil kotak dialog pengaturan **Network** dari **LNX Container**. Pada parameter **IPv4/CIDR**: masukkan alamat IP dan subnetmask yang digunakan oleh *container CentOS 7* yaitu **192.168.169.100/24**. Sedangkan pada bagian **Gateway (IPv4)**:, masukkan alamat IP **192.168.169.254**, seperti terlihat pada gambar berikut:

The screenshot shows the 'Create: LXC Container' dialog box with the 'Network' tab selected. The 'IPv4/CIDR' field is highlighted with a red box and contains '192.168.169.100/24'. The 'Gateway (IPv4)' field is also highlighted with a red box and contains '192.168.169.254'. Other fields include Name: eth0, MAC address: auto, Bridge: vmbr0, VLAN Tag: no VLAN, Rate limit (MB/s): unlimited, and Firewall: unchecked.

Klik tombol **Next** untuk melanjutkan.

Tampil kotak dialog pengaturan **DNS** dari **LNX Container**. Pada parameter **DNS domain**: masukkan nama domain yang digunakan oleh *container CentOS 7*, sebagai contoh menggunakan "**ubg.local**". Sedangkan pada bagian **DNS server1**:, masukkan alamat IP dari **Primary Name Server**, sebagai contoh menggunakan alamat IP **192.168.169.254**, seperti terlihat pada gambar berikut:

Create: LXC Container ✕

General Template Root Disk CPU Memory Network **DNS** Confirm

DNS domain:

DNS servers:

Klik tombol **Next** untuk melanjutkan.

Tampil kotak dialog **Confirm** dari **LNX Container** yang menampilkan ringkasan pengaturan yang telah dilakukan terkait pembuatan *container CentOS 7*, seperti terlihat pada gambar berikut:

Create: LXC Container ✕

General Template Root Disk CPU Memory Network DNS **Confirm**

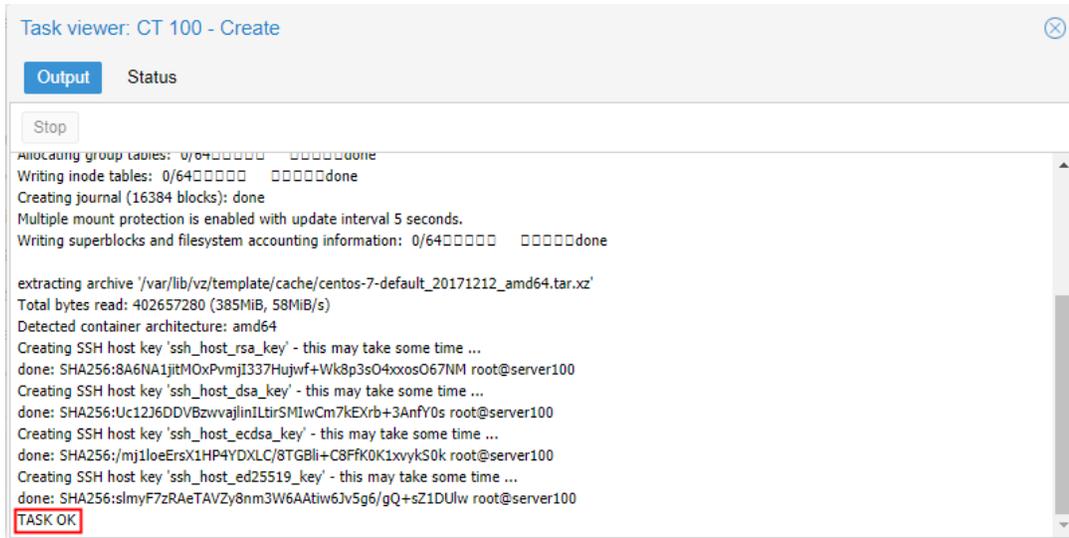
Key ↑	Value
cores	1
hostname	server100.ubg.local
memory	1024
nameserver	192.168.169.254
net0	bridge=vibr0,name=eth0,ip=192.168.169.100/24,gw=192.168.169.254
nodename	pve
ostemplate	local.vztmpl/centos-7-default_20171212_amd64.tar.xz
pool	
rootfs	local-lvm:8
searchdomain	ubg.local
swap	512
vmid	100

Start after created

Advanced **Back** **Finish**

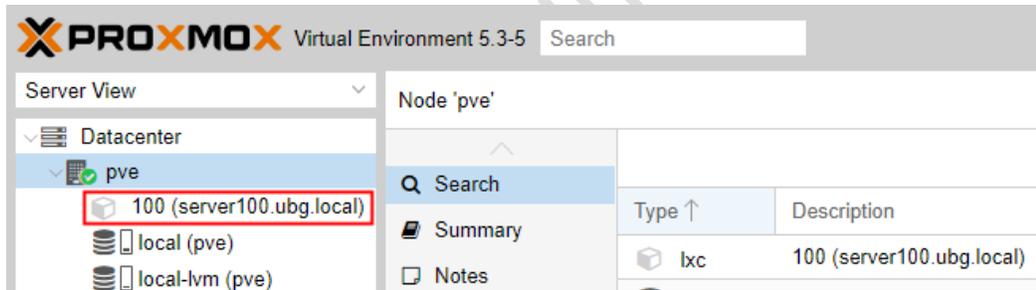
Klik tombol **Finish**.

Tampil kotak dialog **Task viewer: CT 100 – Create**. Tunggu hingga proses pembuatan *container CentOS 7* selesai dibuat dimana ditandai dengan pesan **“TASK OK”** pada bagian **Output** dari kotak dialog **Task viewer: CT 100 – Create**, seperti terlihat pada gambar berikut:

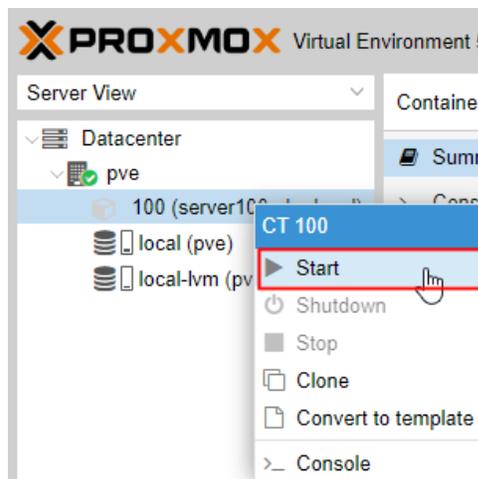


Tutup kotak dialog **Task viewer: CT 100 – Create**.

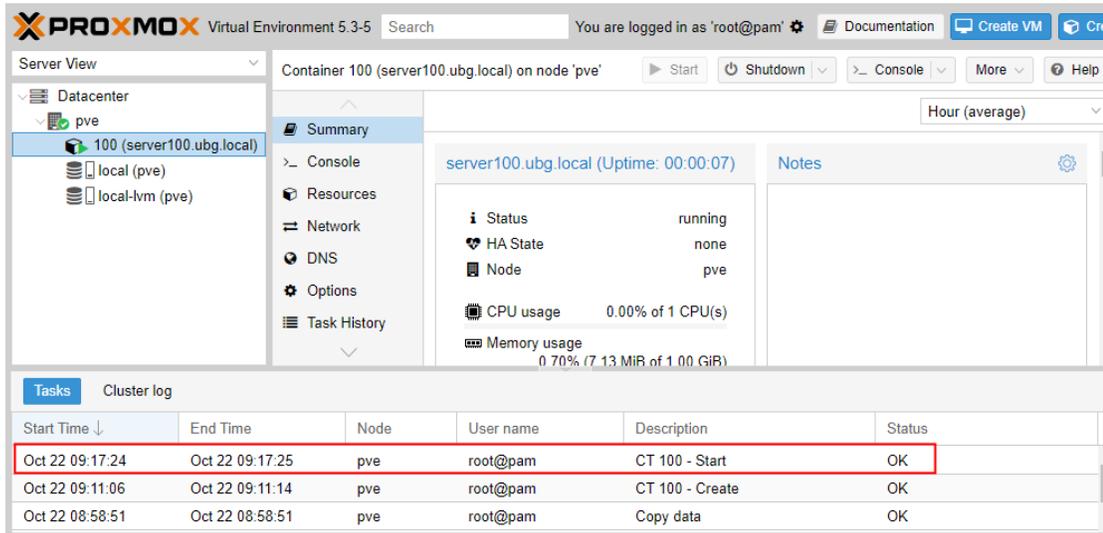
Hasil dari pembuatan *container CentOS 7* dengan ID 100, seperti terlihat pada gambar berikut:



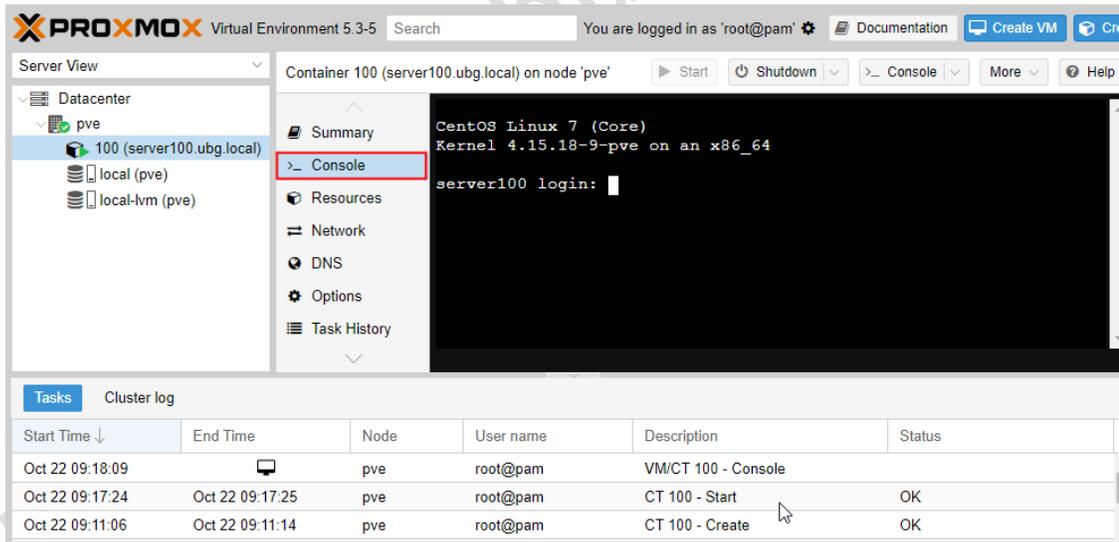
- Untuk menjalankan *Container CentOS 7*, klik kanan pada **“100 (server100.ubg.local)”** di bawah *node “pve”* dari menu **Datacenter** dan pilih **Start**, seperti terlihat pada gambar berikut:



Container CentOS 7 berhasil dijalankan dimana ditandai dengan pesan status **OK** untuk **CT 100 - Start** pada bagian **Tasks** dari **Log Panel**, seperti terlihat pada gambar berikut:



- Untuk mengakses tampilan dari **CT 100**, pilih **Console** pada panel sebelah kanan dari **CT 100 (server100.ubg.local)**, seperti terlihat pada gambar berikut:



Tampil inputan **Server Login** untuk proses otentikasi sebelum pengguna dapat mengakses **Command Line Interface (CLI)** dari **Container CentOS 7**. Masukkan nama login "**root**" pada inputan **Server Login** dan tekan tombol **Enter**.

Tampil inputan **Password:**, masukkan sandi **login** dari user "**root**" yaitu "**12345678**", dan tekan tombol **Enter**. Apabila proses otentikasi login berhasil dilakukan maka akan

tampil *prompt CLI* dari *container CentOS 7* yang ditandai dengan tanda #, seperti terlihat pada gambar berikut:

```
CentOS Linux 7 (Core)
Kernel 4.15.18-9-pve on an x86_64

server100 login: root
Password:
[root@server100 ~]#
```

7. Untuk mematikan **container CentOS 7**, pada **Console** dari *web interface* administrasi **Proxmox** eksekusi perintah “**poweroff**”, seperti terlihat pada gambar berikut:

```
CentOS Linux 7 (Core)
Kernel 4.15.18-9-pve on an x86_64

server100 login: root
Password:
[root@server100 ~]# poweroff
```

Tunggu hingga proses *shutdown* selesai dilakukan.

8. Untuk keluar dari *web interface* administrasi *Proxmox*, klik tombol **Logout** pada bagian *header* paling kanan.

BAB VI

MANAJEMEN USER DAN PERMISSION PADA PROXMOX VE 5.3

Menurut wiki dari *Proxmox*, PVE mendukung berbagai sumber metode otentikasi pengguna meliputi *Linux PAM*, *Proxmox VE Authentication Server*, *LDAP* dan *Microsoft Active Directory*. Akses *granular* dapat didefinisikan dengan menggunakan manajemen *user* dan ijin akses (*permission*) berbasis *role* untuk keseluruhan objek seperti *Virtual Machine (VM)*, *storage*, *node* dan lain-lain.

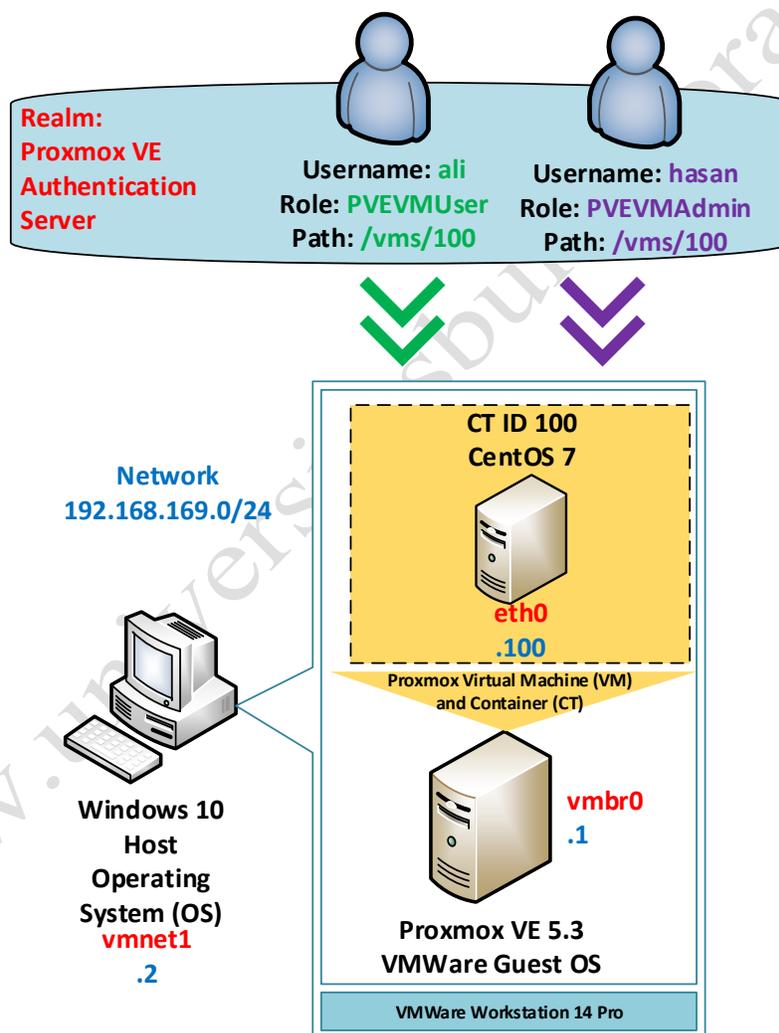
User memerlukan ijin akses yang sesuai untuk dapat melakukan aktivitas seperti melihat, mengubah atau menghapus konfigurasi dari VM. PVE menggunakan sistem manajemen berbasis *role* dan *path*. *Role* merupakan daftar dari hak akses. Terdapat berbagai *role* yang telah didefinisikan oleh PVE, antara lain:

1. Administrator: has all privileges
2. NoAccess: has no privileges (used to forbid access)
3. PVEAdmin: can do most things, but miss rights to modify system settings (*Sys.PowerMgmt*, *Sys.Modify*, *Realm.Allocate*).
4. PVEAuditor: read only access
5. PVEDatastoreAdmin: create and allocate backup space and templates
6. PVEDatastoreUser: allocate backup space and view storage
7. PVEPoolAdmin: allocate pools
8. PVESysAdmin: User ACLs, audit, system console and system logs
9. PVETemplateUser: view and clone templates
10. PVEUserAdmin: user administration
11. PVEVMAdmin: fully administer VMs
12. PVEVMUser: view, backup, config CDROM, VM console, VM power management

Ijin akses diterapkan terhadap objek meliputi *VM*, *storage* atau *pool* dari sumber daya. PVE menggunakan *path* untuk mengamati objek tersebut, sebagai contoh:

1. /nodes/{node} : Access to Proxmox VE server machines
2. /vms : Covers all VMs
3. /vms/{vmid}: Access to specific VMs
4. /storage/{storeid}: Access to a storages
5. /pool/{poolname}: Access to VMs part of a pool
6. /access/groups: Group administration
7. /access/realms/{realmid}: Administrative access to realms

Rancangan *user* yang akan dibuat pada PVE, seperti terlihat pada gambar berikut:

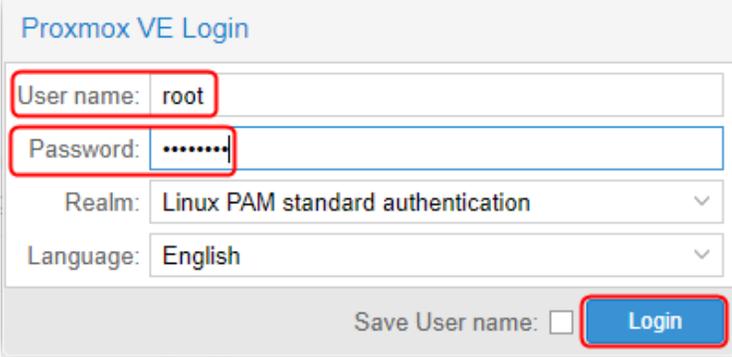


Terdapat 2 (dua) *user* atau pengguna dengan metode otentikasi (realm) **Proxmox VE Authentication Server** yang akan dibuat yaitu **ali** dan **hasan**. *User ali* memiliki *role PVEVMUser*

pada objek **CT ID 100 CentOS 7**. Sedangkan *user hasan* memiliki *role PVEVMAdmin* pada objek **CT ID 100 CentOS 7**.

Adapun langkah-langkah pembuatan *user* dan pengaturan *permission* serta ujicoba berdasarkan rancangan *user* tersebut adalah sebagai berikut:

1. Buka *browser*, sebagai contoh menggunakan **Chrome**. Pada *address bar* dari browser, masukkan URL <https://192.168.169.1:8006>.
2. Tampil kotak dialog otentikasi *Proxmox VE Login*, lengkapi isian “**User name**” dan “**Password**”. Pada isian “*User name*”, masukkan “**root**”. Sedangkan pada isian “*Password*”, masukkan sandi login dari user “*root*” yaitu **12345678**, seperti terlihat pada gambar berikut:



Proxmox VE Login

User name: root

Password: 12345678

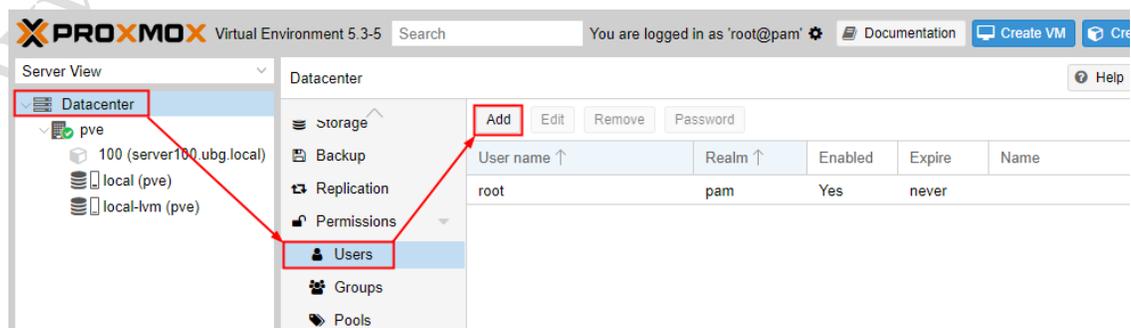
Realm: Linux PAM standard authentication

Language: English

Save User name: Login

Klik tombol **Login**. Pengguna langsung diarahkan ke tampilan halaman *Server View* dari *Proxmox*.

3. Membuat *user* baru yaitu **ali** dan **hasan** dengan mengakses menu **Data Center** pada panel sebelah kiri dan pada panel sebelah kanan memilih submenu **Users** dibawah menu **Permissions** serta memilih tombol **Add**, seperti terlihat pada gambar berikut:



Tampil kotak dialog **Add: User**. Terdapat beberapa parameter yang diatur, seperti terlihat pada gambar berikut:

Penjelasan dari setiap parameter yang dikonfigurasi adalah sebagai berikut:

- User name:**, digunakan untuk mengatur nama login dari pengguna yaitu **“ali”**.
- Realm:**, digunakan untuk menentukan metode otentikasi yang digunakan yaitu **Proxmox VE authentication server**.
- Password:** dan **Confirm Password:** digunakan untuk mengatur sandi login dari user **“ali”** untuk *container* yaitu **“12345678”**.
- First Name:**, digunakan untuk mengatur nama depan dari akun pengguna yang dibuat yaitu **“ali”**.
- Comment:**, digunakan untuk mengatur deskripsi dari akun pengguna yang dibuat yaitu **“ali”**.

Klik tombol **Add** untuk memproses pembuatan *user* **“ali”** maka hasilnya akan terlihat seperti pada gambar berikut:

Datcenter

User name ↑	Realm ↑	Enabled	Expire	Name	Comment
ali	pve	Yes	never	ali	ali
root	pam	Yes	never		

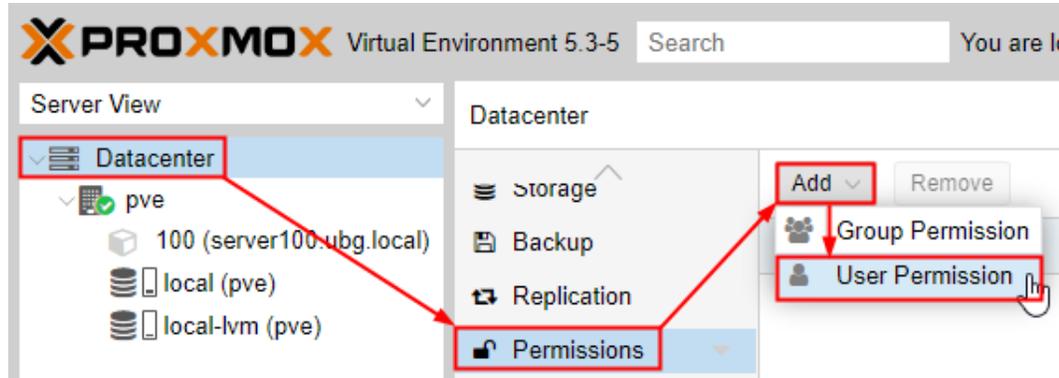
Selanjutnya dengan cara sama, lakukan pula pembuatan user “**hasan**”.

Klik tombol **Add** maka akan tampil kotak dialog **Add: User**. Terdapat beberapa parameter yang diatur, seperti terlihat pada gambar berikut:

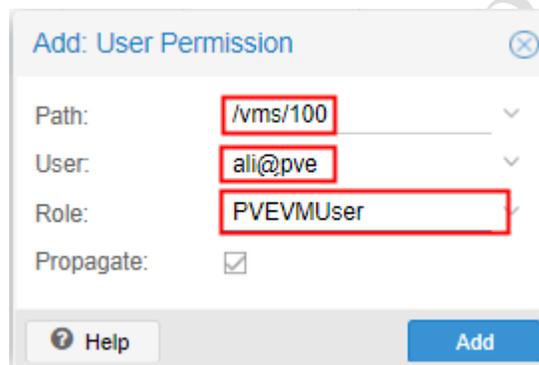
Klik tombol **Add** untuk memproses pembuatan *user* “**hasan**” maka hasilnya akan terlihat seperti pada gambar berikut:

Datacenter						
<input type="button" value="Add"/> <input type="button" value="Edit"/> <input type="button" value="Remove"/> <input type="button" value="Password"/>						
User name ↑	Realm ↑	Enabled	Expire	Name	Comment	
ali	pve	Yes	never	ali	ali	
hasan	pve	Yes	never	hasan	hasan	
root	pam	Yes	never			

- Menambahkan ijin akses agar *user* “**ali**” dapat mengakses objek **CT ID 100 CentOS 7** dengan mengakses menu **Data Center** pada panel sebelah kiri dan pada panel sebelah kanan pilih **Permissions** serta klik tombol **Add > User Permission**, seperti terlihat pada gambar berikut:



Pada kotak dialog **Add: User Permission** yang tampil terdapat beberapa parameter yang diatur, seperti terlihat pada gambar berikut:



Penjelasan dari setiap parameter yang dikonfigurasi adalah sebagai berikut:

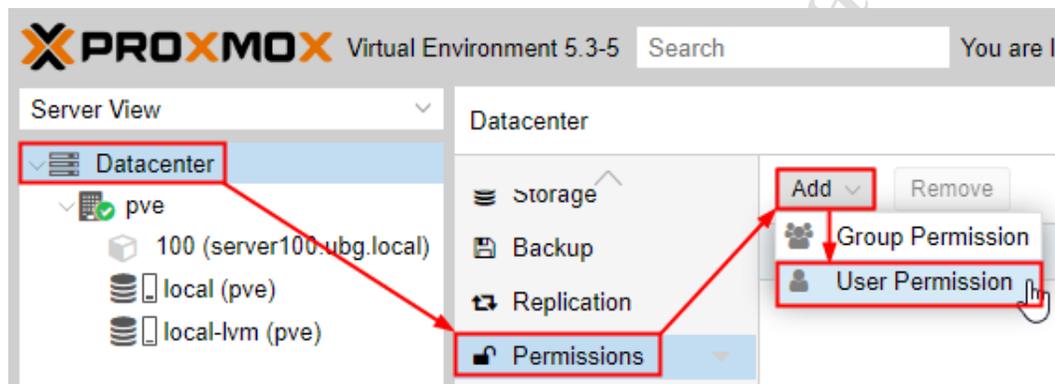
- Path:** digunakan untuk mengatur ijin akses ke objek VM dengan ID tertentu yaitu `"/vms/100"`.
- User:** digunakan untuk menentukan user yang diberikan akses terhadap *path* `"/vms/100"` yaitu `"ali@pve"`.
- Role:** digunakan untuk mengatur *role* yang dialokasikan untuk user `"ali@pve"` yaitu `"PVEVMUser"`. *Role* tersebut memiliki ijin akses untuk melakukan aktivitas *view*, *backup*, *config CDROM*, *VM console*, dan *VM power management*.

Setelah penekanan tombol **Add** maka akan terlihat hasil dari pembuatan ijin akses untuk user `"ali@pve"`, seperti terlihat pada gambar berikut:

Datacenter

Path ↑	User/Group	Role	Propagate
/vms/100	ali@pve	PVEVMUser	true

- Menambahkan izin akses agar user “hasan” dapat mengakses objek **CT ID 100 CentOS 7** dengan mengakses menu **Data Center** pada panel sebelah kiri dan pada panel sebelah kanan pilih **Permissions** serta klik tombol **Add > User Permission**, seperti terlihat pada gambar berikut:



Pada kotak dialog **Add: User Permission** yang tampil terdapat beberapa parameter yang diatur, seperti terlihat pada gambar berikut:

Add: User Permission

Path: /vms/101

User: hasan@pve

Role: PVEVMAAdmin

Propagate:

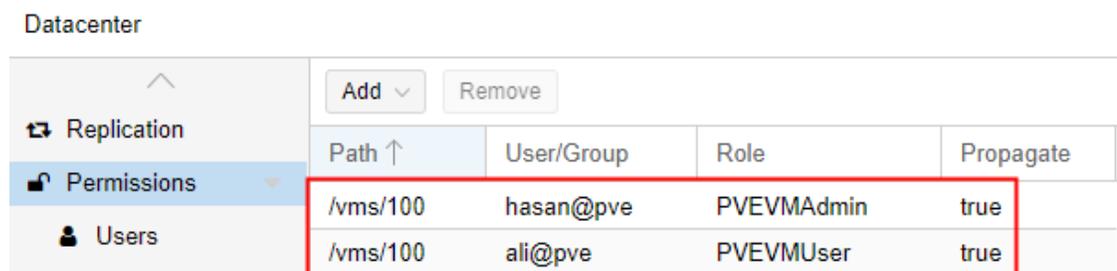
Help Add

Penjelasan dari setiap parameter yang dikonfigurasi adalah sebagai berikut:

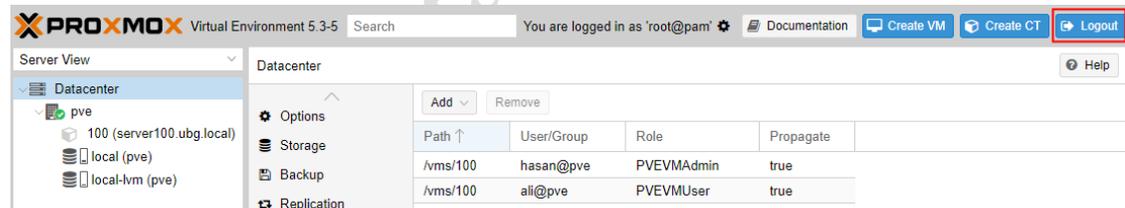
- Path:**, digunakan untuk mengatur izin akses ke objek LXC dengan ID tertentu yaitu “/vms/100”.

- b) **User:** digunakan untuk menentukan user yang diberikan akses terhadap *path* `"/vms/100"` yaitu `"hasan@pve"`.
- c) **Role:** digunakan untuk mengatur *role* yang dialokasikan untuk user `"hasan@pve"` yaitu `"PVEVMAdmin"`. *Role* tersebut memiliki ijin akses untuk melakukan aktivitas manajemen pada *CT ID 100* secara penuh.

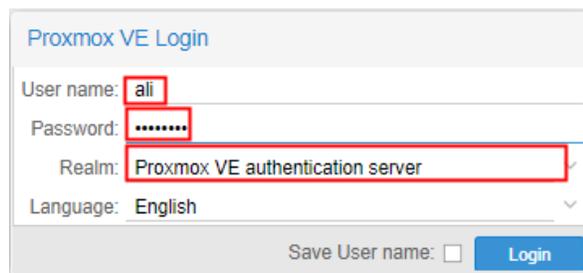
Setelah penekanan tombol **Add** maka akan terlihat hasil dari pembuatan ijin akses untuk user `"hasan@pve"`, seperti terlihat pada gambar berikut:



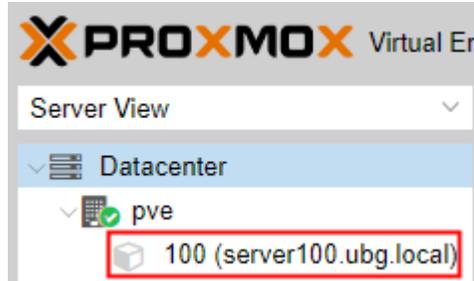
- 6. Klik tombol **Logout** pada bagian pojok kanan atas untuk keluar dari *PVE WebGUI* sebagai user `"admin"` dari, seperti terlihat pada gambar berikut:



- 7. Lakukan login kembali ke *PVE WebGUI* menggunakan user `"ali"` dengan *password* `"123456"` dan *realm* `"Proxmox VE Authentication Server"`, seperti terlihat pada gambar berikut:

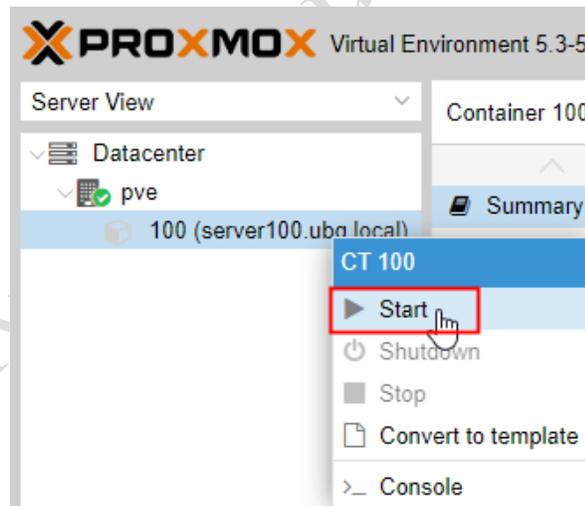


Klik tombol **Login**. Apabila login sukses maka akan tampil *Dashboard PVE*. Pada panel sebelah kiri pilih **Datacenter** > **PVE** maka akan terlihat **CT ID 100** seperti gambar berikut:



Hal ini sesuai dengan izin akses yang diberikan pada user “ali” yaitu hanya dapat mengakses objek **CT ID 100 CentOS 7**.

Selanjutnya jalankan VM Mikrotik CHR dengan cara klik kanan pada “**100 (server100.ubg.local)**” di bawah node “pve” dari menu **Datacenter** dan pilih **Start**, seperti terlihat pada gambar berikut:



Pada bagian **Tasks** dari **Log Panel** memperlihatkan pesan status **OK** untuk **VM 100 – Start** yang menyatakan bahwa VM tersebut berhasil dijalankan, seperti terlihat pada gambar berikut:

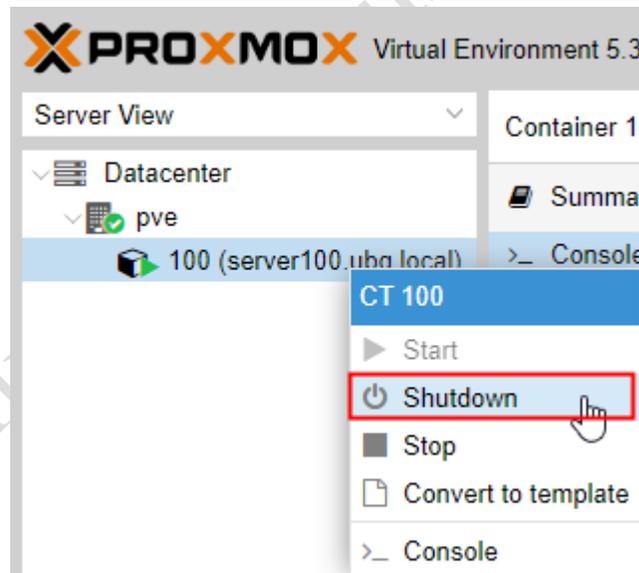
Start Time ↓	End Time	Node	User name	Description	Status
Oct 22 09:44:52	Oct 22 09:44:53	pve	ali@pve	CT 100 - Start	OK

Selanjutnya untuk mengakses tampilan dari **CT 100**, pilih **Console** pada panel sebelah kanan dari **CT 100 (server100.ubg.local)**, seperti terlihat pada gambar berikut:



Tampil inputan **server Login** untuk proses otentikasi sebelum pengguna dapat mengakses **Command Line Interface (CLI)** dari *CentOS 7*.

Selanjutnya lakukan **shutdown CT 100** dengan cara klik kanan pada “**100 (server100.ubg.local)**” di bawah *node “pve”* dari menu **Datacenter** dan pilih **Shutdown**, seperti terlihat pada gambar berikut:



Tampil kotak dialog konfirmasi proses *shutdown*, tekan tombol **Yes**.

Tunggu hingga proses *shutdown* selesai dilakukan dan **Logout** sebagai user “**ali**” dari *PVE WebGUI*.

8. Lakukan login kembali ke *PVE WebGUI* menggunakan user “**hasan**” dengan *password* “**123456**” dan *realm* “**Proxmox VE Authentication Server**”, seperti terlihat pada gambar berikut:

Proxmox VE Login

User name:

Password:

Realm:

Language:

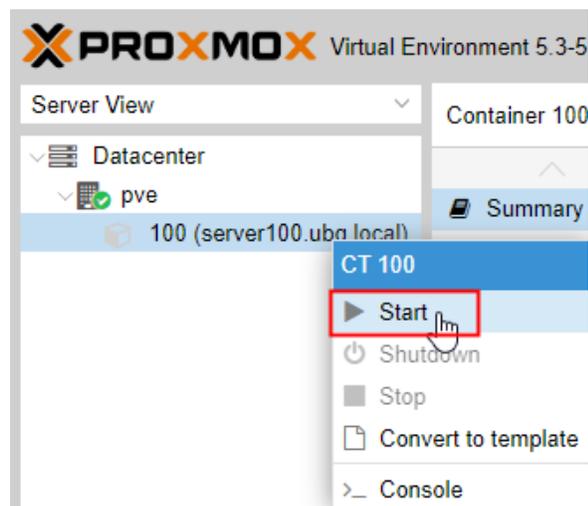
Save User name:

Klik tombol **Login**. Apabila login sukses maka akan tampil *Dashboard PVE*. Pada panel sebelah kiri pilih **Datacenter** > **PVE** maka akan terlihat **CT ID 100** seperti gambar berikut:



Hal ini sesuai dengan ijin akses yang diberikan pada user “**hasan**” yaitu hanya dapat mengakses objek **CT ID 100 CentOS 7**.

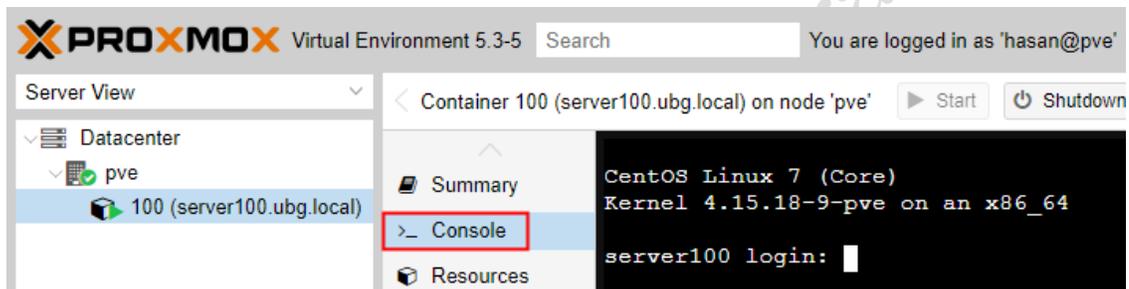
Selanjutnya jalankan *CT ID 100* dengan cara klik kanan pada “**100 (server100.ubg.local)**” di bawah *node “pve”* dari menu **Datacenter** dan pilih **Start**, seperti terlihat pada gambar berikut:



Pada bagian **Tasks** dari **Log Panel** memperlihatkan pesan status **OK** untuk **CT 100 – Start** yang menyatakan bahwa CT tersebut berhasil dijalankan, seperti terlihat pada gambar berikut:

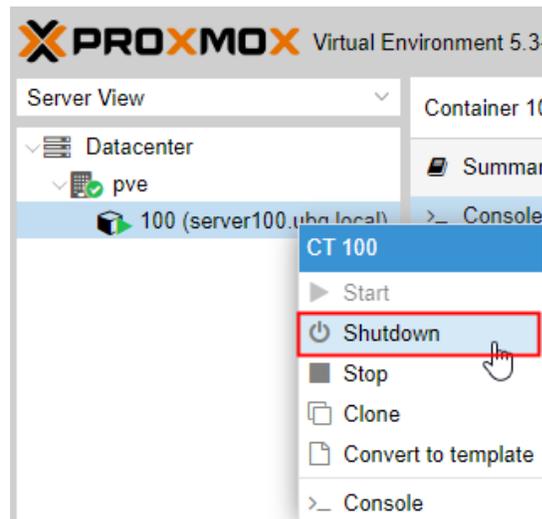
Start Time ↓	End Time	Node	User name	Description	Status
Oct 22 09:51:12	Oct 22 09:51:13	pve	hasan@pve	CT 100 - Start	OK

Selanjutnya untuk mengakses tampilan dari **CT 100**, pilih **Console** pada panel sebelah kanan dari **CT 100 (server100.ubg.local)**, seperti terlihat pada gambar berikut:



Tampil inputan **server Login** untuk proses otentikasi sebelum pengguna dapat mengakses **Command Line Interface (CLI)** dari **CentOS 7**.

Selanjutnya lakukan **shutdown CT 100** dengan cara klik kanan pada **“100 (server100.ubg.local)”** di bawah **node “pve”** dari menu **Datacenter** dan pilih **Shutdown**, seperti terlihat pada gambar berikut:



Tampil kotak dialog konfirmasi proses *shutdown*, tekan tombol **Yes**.

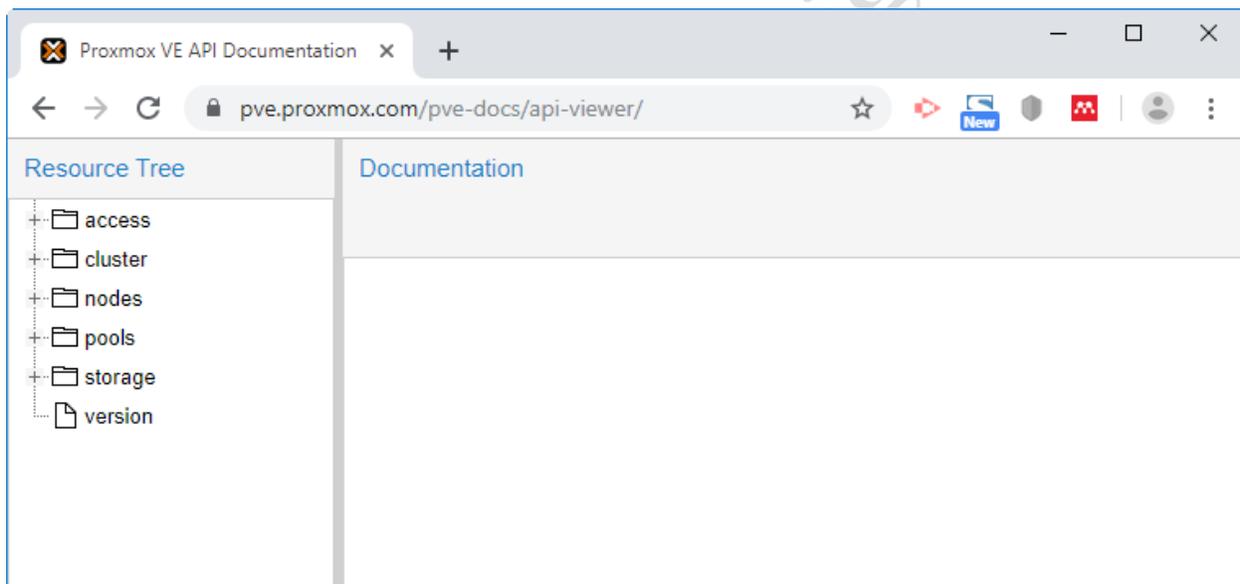
Tunggu hingga proses *shutdown* selesai dilakukan dan **Logout** sebagai user “**hasan**” dari *PVE WebGUI*.

www.universitasbumigora.ac.id

BAB VII

PROXMOX VE 5.3 APPLICATION PROGRAMMING INTERFACE (API)

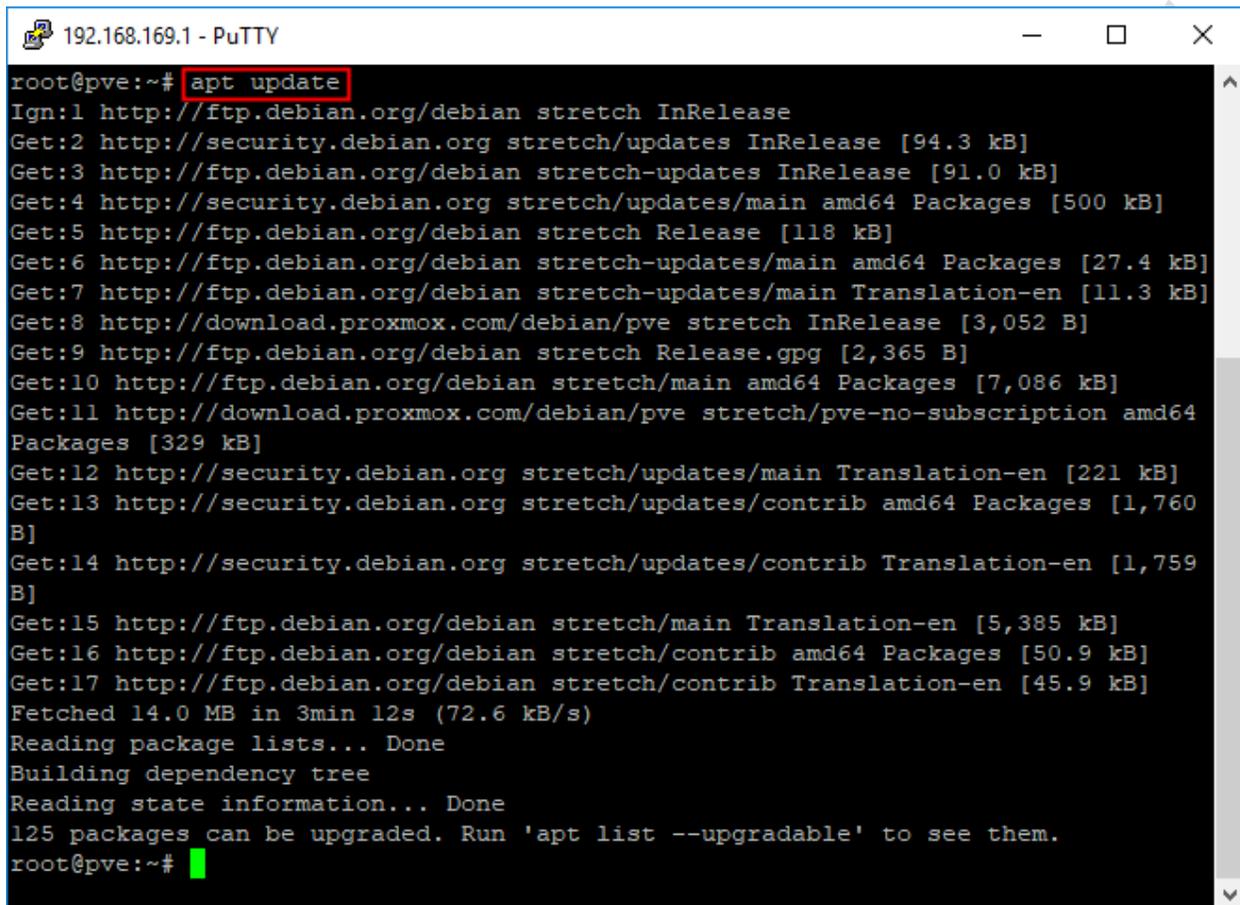
Menurut [wiki](#), Proxmox VE menggunakan *Representational State Transfer (REST) Application Programming Interface (API)* dengan format data *JavaScript Object Notation (JSON)* yang dapat digunakan untuk mengontrol server PVE baik melalui *command line* maupun *Web Graphical User Interface (GUI)*. Secara formal API PVE didefinisikan menggunakan *JSON schema*. Dokumentasi API dari PVE yang memuat parameter dan *return value* dapat diakses pada alamat <https://pve.proxmox.com/pve-docs/api-viewer/>, seperti terlihat pada gambar berikut:



API menggunakan protokol HTTPS dan *port* 8006 dengan alamat akses yaitu <https://ip.dari.server.pve:8006/api2/json/>.

PVE menggunakan otentikasi berbasis *token* sehingga keseluruhan permintaan ke API memerlukan *token* yang disimpan di dalam *cookie* dan disebut dengan **ticket**. Selain itu *API request* harus memuat *Cross-Site Request Forgery (CSRF) prevention token*. Mekanisme otentikasi ini serupa dengan ketika mengakses *Web GUI PVE* yang memerlukan akun *username* dan *password* serta *realm*. Utilitas **curl** yang dieksekusi dari *command line* dapat digunakan untuk

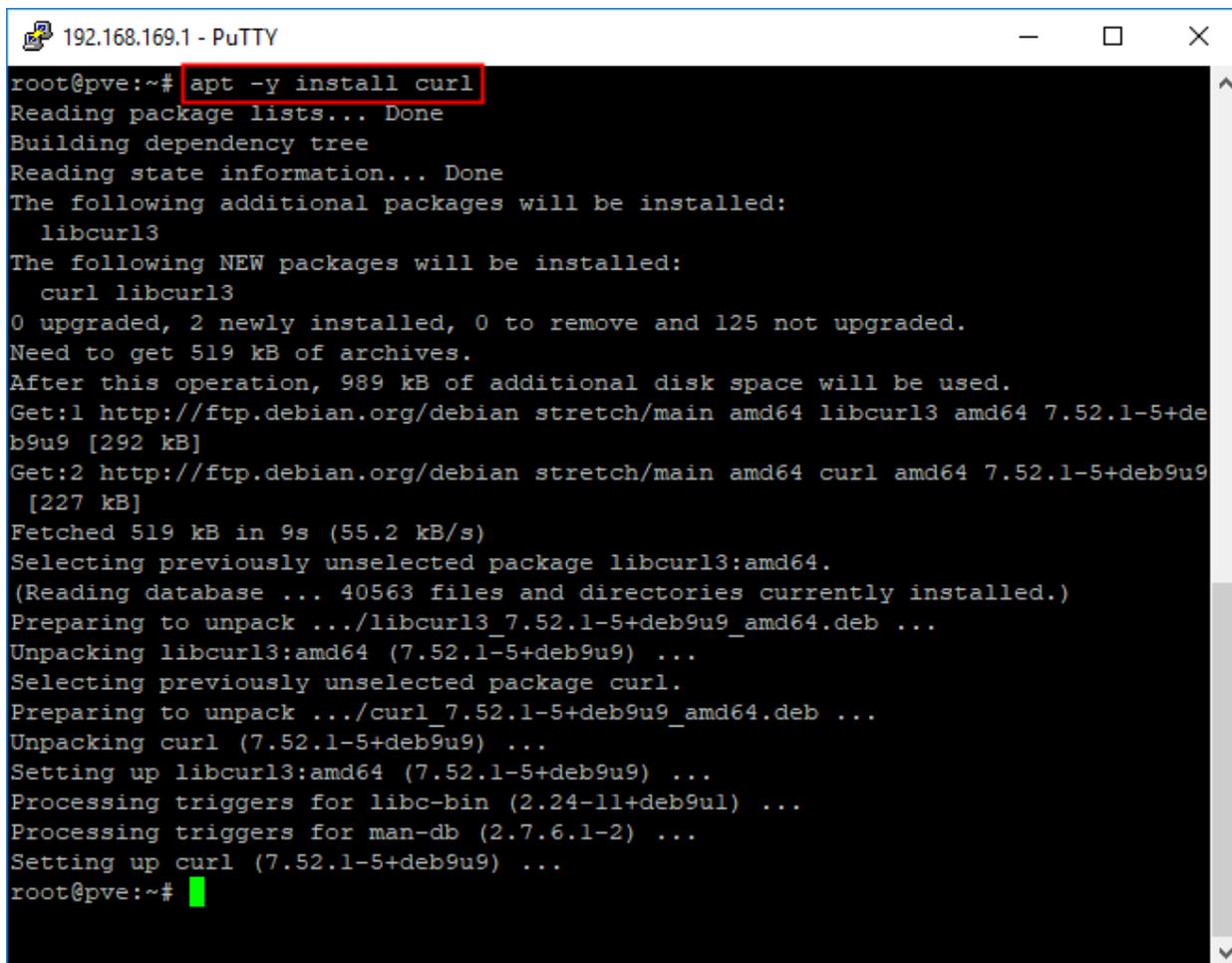
memperoleh *ticket* baru dan *CSRF prevention token* dari server PVE. Utilitas **curl** ini belum tersedia secara *default* pada server PVE. Sebelum melakukan instalasi paket **curl** tersebut, lakukan pembaharuan pada server PVE dengan mengeksekusi perintah “**apt update**”, seperti terlihat pada gambar berikut:



```
192.168.169.1 - PuTTY
root@pve:~# apt update
Ign:1 http://ftp.debian.org/debian stretch InRelease
Get:2 http://security.debian.org stretch/updates InRelease [94.3 kB]
Get:3 http://ftp.debian.org/debian stretch-updates InRelease [91.0 kB]
Get:4 http://security.debian.org stretch/updates/main amd64 Packages [500 kB]
Get:5 http://ftp.debian.org/debian stretch Release [118 kB]
Get:6 http://ftp.debian.org/debian stretch-updates/main amd64 Packages [27.4 kB]
Get:7 http://ftp.debian.org/debian stretch-updates/main Translation-en [11.3 kB]
Get:8 http://download.proxmox.com/debian/pve stretch InRelease [3,052 B]
Get:9 http://ftp.debian.org/debian stretch Release.gpg [2,365 B]
Get:10 http://ftp.debian.org/debian stretch/main amd64 Packages [7,086 kB]
Get:11 http://download.proxmox.com/debian/pve stretch/pve-no-subscription amd64 Packages [329 kB]
Get:12 http://security.debian.org stretch/updates/main Translation-en [221 kB]
Get:13 http://security.debian.org stretch/updates/contrib amd64 Packages [1,760 B]
Get:14 http://security.debian.org stretch/updates/contrib Translation-en [1,759 B]
Get:15 http://ftp.debian.org/debian stretch/main Translation-en [5,385 kB]
Get:16 http://ftp.debian.org/debian stretch/contrib amd64 Packages [50.9 kB]
Get:17 http://ftp.debian.org/debian stretch/contrib Translation-en [45.9 kB]
Fetched 14.0 MB in 3min 12s (72.6 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
125 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@pve:~#
```

Tunggu hingga proses pembaharuan selesai dilakukan.

Selanjutnya lakukan instalasi paket **curl** dengan mengeksekusi perintah “**apt -y install curl**”, seperti terlihat pada gambar berikut:



```

192.168.169.1 - PuTTY
root@pve:~# apt -y install curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libcurl3
The following NEW packages will be installed:
  curl libcurl3
0 upgraded, 2 newly installed, 0 to remove and 125 not upgraded.
Need to get 519 kB of archives.
After this operation, 989 kB of additional disk space will be used.
Get:1 http://ftp.debian.org/debian stretch/main amd64 libcurl3 amd64 7.52.1-5+deb9u9 [292 kB]
Get:2 http://ftp.debian.org/debian stretch/main amd64 curl amd64 7.52.1-5+deb9u9 [227 kB]
Fetched 519 kB in 9s (55.2 kB/s)
Selecting previously unselected package libcurl3:amd64.
(Reading database ... 40563 files and directories currently installed.)
Preparing to unpack .../libcurl3_7.52.1-5+deb9u9_amd64.deb ...
Unpacking libcurl3:amd64 (7.52.1-5+deb9u9) ...
Selecting previously unselected package curl.
Preparing to unpack .../curl_7.52.1-5+deb9u9_amd64.deb ...
Unpacking curl (7.52.1-5+deb9u9) ...
Setting up libcurl3:amd64 (7.52.1-5+deb9u9) ...
Processing triggers for libc-bin (2.24-11+deb9u1) ...
Processing triggers for man-db (2.7.6.1-2) ...
Setting up curl (7.52.1-5+deb9u9) ...
root@pve:~#

```

Contoh penggunaan utilitas `curl` yang dieksekusi melalui *command line* untuk memperoleh *ticket* baru dan *CSRF prevention token* pada server PVE dengan alamat IP **192.168.169.1** serta *login* sebagai user "**root**" dengan *realm* "**pam**" dan *password* "**12345678**" adalah sebagai berikut:

```
# curl -k -d "username=root@pam&password=12345678"
https://192.168.169.1:8006/api2/json/access/ticket
```

Hasil dari eksekusi perintah tersebut seperti terlihat pada gambar berikut:

```
192.168.169.1 - PuTTY
root@pve:~# curl -k -d "username=root@pam&password=12345678" https://192.168.169.1:8006/api2/json/access/ticket
{"data":{"CSRFPreventionToken":"5DB17906:WIs01JcSVjsOX0afVedukvXx/GQ","ticket":"PVE:root@pam:5DB17906:YpF4z1j62lm
vhH4xLk7pOqske5LdH8DAWYSpoaWgAxXWHzuIusB5RPoEg2+WFO9SDmJ81VY71jh4EnCd0mn15IAAvRFzrrV70PoKK8VjUMsu7sbKP7wf+Z/zyVd8
9UmfeaQJ/MfsZxIG6hAWyEsZC404iYUo7CGeD6eBiEAprMetZ3qnITEvFUhgC7zWNovhcbKRyO01fCjTycWW7f6BLv4v3MpuCrzAYvL4nyrfp4oYTZ
FEUx6xxv5Qm945pzAa03s7ylJ2coMd/B9SUHwi2jgnEQ9kLfQn0rfgYSMGbUK+exDbaBgHbBv0n9PaQUtmwMGMWIKcWr/NFp5JxQVlw==","cap":{"
"vms":{"VM.Config.HWType":1,"VM.PowerMgmt":1,"VM.Config.CPU":1,"VM.Console":1,"VM.Snapshot.Rollback":1,"VM.Config.
CDROM":1,"VM.Backup":1,"VM.Migrate":1,"Permissions.Modify":1,"VM.Config.Network":1,"VM.Clone":1,"VM.Config.Options
":1,"VM.Audit":1,"VM.Snapshot":1,"VM.Monitor":1,"VM.Config.Memory":1,"VM.Allocate":1,"VM.Config.Disk":1},"nodes":{"
"Sys.Syslog":1,"Sys.Modify":1,"Permissions.Modify":1,"Sys.PowerMgmt":1,"Sys.Console":1,"Sys.Audit":1},"access":{"P
ermissions.Modify":1,"Group.Allocate":1,"User.Modify":1},"storage":{"Datastore.Allocate":1,"Datastore.Audit":1,"Da
tastore.AllocateTemplate":1,"Permissions.Modify":1,"Datastore.AllocateSpace":1},"dc":{"Sys.Audit":1}},"username":
"root@pam"}}root@pve:~#
```

Ticket memiliki masa kadaluarsa 2 jam sehingga diperlukan permintaan ulang ketika telah habis masa berlakunya.

PVE menyediakan utilitas *command line* **pvesh** yang dapat digunakan untuk mengakses REST API. Hasil eksekusi dari perintah **pvesh** pada terminal, seperti terlihat pada gambar berikut:

```
root@pve:~# pvesh
ERROR: no command specified
USAGE: pvesh <COMMAND> [ARGS] [OPTIONS]

pvesh create <api_path> [OPTIONS] [FORMAT_OPTIONS]
pvesh delete <api_path> [OPTIONS] [FORMAT_OPTIONS]
pvesh get <api_path> [OPTIONS] [FORMAT_OPTIONS]
pvesh ls <api_path> [OPTIONS] [FORMAT_OPTIONS]
pvesh set <api_path> [OPTIONS] [FORMAT_OPTIONS]
pvesh usage <api_path> [OPTIONS]

pvesh help [<extra-args>] [OPTIONS]
```

Terlihat bahwa utilitas **pvesh** memiliki 7 (tujuh) *command* yaitu **create**, **delete**, **get**, **ls**, **set** dan **usage** serta **help**.

- *Command create* digunakan untuk memanggil API POST sehingga dapat membuat sumber daya baru atau mengatur nilai pada objek yang belum diatur sebelumnya pada *server* PVE.
- *Command delete* digunakan untuk memanggil API DELETE sehingga dapat menghapus sumber daya tertentu pada *server* PVE.
- *Command get* digunakan untuk memanggil API GET sehingga dapat membaca atau mengambil informasi terkait objek pada *server* PVE.
- *Command ls* digunakan untuk menampilkan daftar *child object* pada API PATH tertentu.

- *Command set* digunakan untuk memanggil API PUT sehingga dapat memperbaharui nilai dari sumber daya atau objek pada *server* PVE.
- *Command usage* digunakan untuk menampilkan informasi penggunaan API dari API PATH tertentu.
- *Command help* digunakan untuk menampilkan informasi terkait tata cara penggunaan perintah tertentu dan argumen serta *options* yang dimiliki oleh perintah tersebut.

Gambar berikut memperlihatkan cuplikan *output* dari hasil eksekusi utilitas **pvesh help** pada *terminal*:

```
root@pve:~# pvesh help
USAGE: pvesh create <api_path> [OPTIONS] [FORMAT_OPTIONS]

Call API POST on <api_path>.

<api_path> <string>
           API path.

-noproxy  <boolean>
           Disable automatic proxying.

USAGE: pvesh delete <api_path> [OPTIONS] [FORMAT_OPTIONS]

Call API DELETE on <api_path>.

<api_path> <string>
           API path.

-noproxy  <boolean>
           Disable automatic proxying.

USAGE: pvesh get <api_path> [OPTIONS] [FORMAT_OPTIONS]

Call API GET on <api_path>.

<api_path> <string>
           API path.

-noproxy  <boolean>
           Disable automatic proxying.
```

Adapun contoh-contoh penggunaan dari perintah **pvesh** untuk manajemen objek pada PVE adalah sebagai berikut:

A. Manajemen Container Template

Adapun langkah-langkah manajemen *container template* adalah sebagai berikut:

1. Menampilkan informasi *storage* atau media penyimpanan yang dimiliki oleh *server* PVE menggunakan sintak penulisan:

```
# pvesh get /nodes/{node}/storage
```

Ubah {node} dengan *hostname* dari *server proxmox* yaitu **pve** sehingga menjadi:

```
# pvesh get /nodes/pve/storage
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
root@pve:~# pvesh get /nodes/pve/storage
```

content	storage	type	active	avail	enabled	shared	total	used	used_fraction
images,rootdir	local-lvm	lvmthin	1	18.28 GiB	1	0	18.87 GiB	608.71 MiB	3.15%
vztmpl,iso,backup	local	dir	1	7.33 GiB	1	0	9.53 GiB	1.70 GiB	17.80%

Output pada kolom **storage** menunjukkan bahwa *server* PVE memiliki 2 (dua) *storage* yaitu **local-lvm** dan **local**.

2. Menampilkan informasi *container template* yang terdapat pada *server* PVE menggunakan sintak penulisan:

```
# pvesh get /nodes/{node}/storage/{storage}/content
```

Ubah {node} dengan *hostname* dari *server proxmox* yaitu **pve** dan {storage} dengan nama pengenal dari media penyimpanan yaitu **local** sehingga menjadi:

```
# pvesh get /nodes/pve/storage/local/content
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
root@pve:~# pvesh get /nodes/pve/storage/local/content
```

format	size	valid	parent	used	vmid
txz	65.96 MiB	local:vztmpl/centos-7-default_20171212_amd64.tar.xz			

Output pada kolom **valid** menunjukkan bahwa terdapat *container template* dengan nama "**centos-7-default_20171212_amd64.tar.xz**" di *storage* **local**.

3. Mengunggah *container template* ke *server* PVE dengan sintak penulisan:

```
# pvesh create /nodes/{node}/storage/{storage}/upload
```

Ubah {node} dengan *hostname* dari *server proxmox* yaitu **pve** dan {storage} dengan nama pengenal dari media penyimpanan yaitu **local** sehingga menjadi:

```
# pvesh create /nodes/pve/storage/local/upload
```

Selain itu diperlukan pengaturan 3 (tiga) parameter antara lain:

- `content` digunakan untuk menentukan jenis konten yang akan diunggah yaitu **vztmpl** untuk *container template*.
- `filename` digunakan untuk mengatur nama *file* yang akan dibuat, sebagai contoh **centos-8-default_20191016_amd64.tar.xz**.
- `tmpfilename` digunakan untuk mengatur nama file sumber yang akan diunggah sebagai contoh **CentOS 8** dengan nama file **centos-8-default_20191016_amd64.tar.xz**.

sehingga menjadi:

```
# pvesh create /nodes/pve/storage/local/upload --content
vztmpl --filename centos-8-default_20191016_amd64.tar.xz --
tmpfilename centos-8-default_20191016_amd64.tar.xz
```

PERHATIAN:

Sebelum melakukan proses pengunggahan maka pastikan terlebih dahulu pada *home* direktori dari *user root* telah terdapat *file container template* yang akan diunggah ke *storage local* dari *server PVE* yaitu "**centos-8-default_20191016_amd64.tar.xz**", seperti terlihat pada gambar berikut:

```
root@pve:~# pwd
/root
root@pve:~# ls
centos-8-default_20191016_amd64.tar.xz
```

Apabila belum tersedia maka lakukan pengunggahan file tersebut terlebih dahulu menggunakan aplikasi **WinSCP** atau *file transfer* lainnya. Sebagai alternatif dapat pula diunduh secara langsung dari Internet dengan mengeksekusi perintah:

```
wget http://download.proxmox.com/images/system/centos-8-default_20191016_amd64.tar.xz
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
root@pve:~# pvsh create /nodes/pve/storage/local/upload --content vztmpl --filename centos-8-default_20191016_amd64.tar.xz --tmpfilename centos-8-default_20191016_amd64.tar.xz
starting file import from: centos-8-default_20191016_amd64.tar.xz
target node: pve
target file: /var/lib/vz/template/cache/centos-8-default_20191016_amd64.tar.xz
file size is: 106244064
command: cp -- centos-8-default_20191016_amd64.tar.xz /var/lib/vz/template/cache/centos-8-default_20191016_amd64.tar.xz
finished file import successfully
UPID:pve:000029EA:0007B36B:5DB1A5A7:imgcopy::root@pam:
```

File *container template CentOS 8* yang telah berhasil diunggah ke *storage local* dari server **pve** dapat diverifikasi menggunakan perintah:

```
# pvsh get /nodes/pve/storage/local/content
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
root@pve:~# pvsh get /nodes/pve/storage/local/content
```

format	size	volid	parent	used	vmid
txz	65.96 MiB	local:vztmpl/centos-7-default_20171212_amd64.tar.xz			
txz	101.32 MiB	local:vztmpl/centos-8-default_20191016_amd64.tar.xz			

4. Menghapus *container template* tertentu di server PVE dengan sintak penulisan:

```
# pvsh delete /nodes/{node}/storage/{storage}/content/{volume}
```

Ubah {node} dengan *hostname* dari server *proxmox* yaitu **pve** dan {storage} dengan nama pengenalan dari media penyimpanan yaitu **local** serta {volume} berupa *volume identifier* yaitu **local:vztmpl/centos-8-default_20191016_amd64.tar.xz**, sehingga menjadi:

```
# pvsh delete /nodes/pve/storage/local/content/local:vztmpl/centos-8-default_20191016_amd64.tar.xz
```

Verifikasi terhadap *file container template CentOS 8* yang telah dihapus dari *storage local* di server **pve** dilakukan dengan mengeksekusi perintah:

```
# pvsh get /nodes/pve/storage/local/content
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
root@pve:~# pvsh get /nodes/pve/storage/local/content
```

format	size	volid	parent	used	vmid
txz	65.96 MiB	local:vztmpl/centos-7-default_20171212_amd64.tar.xz			

B. Manajemen Linux Container (LXC)

Adapun langkah-langkah manajemen LXC adalah sebagai berikut:

1. Menampilkan informasi LXC yang terdapat di *server* PVE menggunakan sintak penulisan:

```
# pvesh get /nodes/{node}/lxc
```

Ubah {node} dengan *hostname* dari *server proxmox* yaitu **pve** sehingga menjadi:

```
# pvesh get /nodes/pve/lxc
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
root@pve:~# pvesh get /nodes/pve/lxc
```

status	vmid	cpus	maxdisk	maxmem	maxswap	name	uptime
stopped	100	1	8.00 GiB	1.00 GiB	512.00 MiB	server100.ubg.local	

Output menunjukkan bahwa *server* PVE memiliki 1 (satu) *container* yaitu dengan **vmid 100**.

2. Membuat *container* baru di *server* PVE dengan sintak penulisan:

```
# pvesh create /nodes/{node}/lxc
```

Ubah {node} dengan *hostname* dari *server proxmox* yaitu **pve** sehingga menjadi:

```
# pvesh create /nodes/pve/lxc
```

Selain itu diperlukan pengaturan beberapa parameter antara lain:

- a) `vmid` digunakan untuk mengatur ID dari VM dan bersifat unik sebagai contoh **101**.
- b) `hostname` digunakan untuk mengatur nama komputer dan nama domain dari *container*, sebagai contoh **server101.ubg.local**.
- c) `password` digunakan untuk mengatur sandi login bagi *user root* dari *container*, sebagai contoh **rahasia**.
- d) `ostemplate` berfungsi untuk menentukan *container template* yang akan digunakan, sebagai contoh **centos-7-default_20171212_amd64.tar.xz** yang terdapat di *storage local*.
- e) `storage` berfungsi untuk mengatur pool media penyimpanan yang digunakan oleh *container* meliputi **local** atau **local-lvm**, sebagai contoh **local-lvm**.

- f) `rootfs` digunakan untuk mengatur volume sebagai *container root* dan ukuran dari hardisk yang dialokasikan untuk *container*, sebagai contoh **5** dalam satuan *Gigabyte*.
- g) `memory` digunakan untuk mengatur ukuran *Random Access Memory (RAM)* dari dalam satuan *Megabyte (MB)*, sebagai contoh **512**.
- h) `swap` digunakan untuk mengatur ukuran *swap* dari VM dalam satuan *Megabyte (MB)*, sebagai contoh **512**.
- i) `net[n]` digunakan untuk mengatur *interface* jaringan dari *container* dimana `n` adalah nomor *interface*, sebagai contoh **net0** untuk *interface* jaringan pertama. Parameter ini memiliki format penulisan sebagai berikut:
- `name=<string>, ip=<(IPv4/CIDR|dhcp|manual)>, gw=<Gateway IPv4>, bridge=<bridge>`.
- `name` digunakan untuk mengatur nama *interface* jaringan, sebagai contoh **eth0**.
 - `ip` digunakan untuk menentukan metode alokasi pengalamanan IP pada *interface* meliputi **dhcp**, **manual** atau dicantumkan secara statik dengan format penulisan **IPv4/CIDR**, sebagai contoh **192.168.169.101/24**.
 - `gw` digunakan untuk mengatur alamat *default gateway* atau *router* sehingga *container* dapat berkomunikasi ke beda *network*, sebagai contoh **192.168.169.254**.
 - `bridge` digunakan untuk mengatur *bridge* yang digunakan oleh *interface* jaringan, sebagai contoh **vbr0**.
- j) `nameserver` digunakan untuk mengatur alamat IP dari server DNS, sebagai contoh **192.168.169.254**.
- k) `searchdomain` digunakan untuk mengatur nama domain pencarian DNS, sebagai contoh **ubg.local**.

sehingga menjadi:

```
#pvesh create /nodes/pve/lxc \
-vmid 101 -hostname server101.ubg.local -password rahasia \
-ostemplate local:vztmpl/centos-7-default_20171212_amd64.tar.xz \
-storage local-lvm -rootfs 5 \
-memory 512 -swap 512 \
-net0 name=eth0,ip=192.168.169.101/24,gw=192.168.169.254,bridge=vbr0 \
-nameserver 192.168.169.254 -searchdomain ubg.local
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
root@pve:~# pvesh create /nodes/pve/lxc \
> -vmid 101 -hostname server101.ubg.local -password rahasia \
> -ostemplate local:vztmpl/centos-7-default_20171212_amd64.tar.xz \
> -storage local-lvm -rootfs 5 \
> -memory 512 -swap 512 \
> -net0 name=eth0,ip=192.168.169.101/24,gw=192.168.169.254,bridge=vbr0 \
> -nameserver 192.168.169.254 -searchdomain ubg.local
Using default stripesize 64.00 KiB.
Logical volume "vm-101-disk-0" created.
mke2fs 1.43.4 (31-Jan-2017)
Discarding device blocks: done
Creating filesystem with 1310720 4k blocks and 327680 inodes
Filesystem UUID: cb60945b-b956-4728-a276-5c2599ac7747
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Multiple mount protection is enabled with update interval 5 seconds.
Writing superblocks and filesystem accounting information: done

extracting archive '/var/lib/vz/template/cache/centos-7-default_20171212_amd64.tar.xz'
Total bytes read: 402657280 (385MiB, 12MiB/s)
Detected container architecture: amd64
Creating SSH host key 'ssh_host_dsa_key' - this may take some time ...
done: SHA256:CW7DNmc+AKEZBArRgpYZOdrin4m5YTxGLU57coTDKQs root@server101
Creating SSH host key 'ssh_host_rsa_key' - this may take some time ...
done: SHA256:LCRrzpmc5mmzRVCZdO6YwjV+RN37dDUtcywxr6eVqYc root@server101
Creating SSH host key 'ssh_host_ecdsa_key' - this may take some time ...
done: SHA256:KP7wT//u7H7VrF0H1BZj0sIsYOV6+DaKs9sFjg5KOHo root@server101
Creating SSH host key 'ssh_host_ed25519_key' - this may take some time ...
done: SHA256:FGtXrolB6uPyyYb7L6rWBvLjZiB4LcfkfSWDHqm/MG4 root@server101
UPID:pve:00000D58:0001ED52:5DB3AE24:vzcreate:101:root@pam:
```

3. Memverifikasi hasil dari pembuatan *container* baru yang memiliki ID **101** dengan mengeksekusi perintah berikut:

```
# pvesh get /nodes/pve/lxc
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
root@pve:~# pvesh get /nodes/pve/lxc
```

status	vmid	cpus	maxdisk	maxmem	maxswap	name	uptime
stopped	100	1	8.00 GiB	1.00 GiB	512.00 MiB	server100.ubg.local	
stopped	101	1	5.00 GiB	512.00 MiB	512.00 MiB	server101.ubg.local	

Output menunjukkan bahwa *container* dengan **vmid 101** telah berhasil dibuat.

- Menghapus *container* tertentu di server PVE dengan sintak penulisan:

```
# pvesh delete /nodes/{node}/lxc/{vmid}
```

Ubah {node} dengan *hostname* dari *server proxmox* yaitu **pve** dan {vmid} dengan ID dari VM yang akan dihapus sebagai contoh **101**, sehingga menjadi:

```
# pvesh delete /nodes/pve/lxc/101
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
root@pve:~# pvesh delete /nodes/pve/lxc/101
Logical volume "vm-101-disk-0" successfully removed
UPID:pve:000024D4:0007EF8C:5DB2C9AD:vzdestroy:101:root@pam:
```

- Memverifikasi hasil dari penghapusan *container* yang memiliki ID **101** dengan mengeksekusi perintah berikut:

```
# pvesh get /nodes/pve/lxc
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
root@pve:~# pvesh get /nodes/pve/lxc
```

status	vmid	cpus	maxdisk	maxmem	maxswap	name	uptime
stopped	100	1	8.00 GiB	1.00 GiB	512.00 MiB	server100.ubg.local	

Output menunjukkan bahwa tidak terdapat *container* dengan **vmid 101** sehingga *container* tersebut telah berhasil dihapus.

C. Manajemen User

Adapun langkah-langkah manajemen *user* adalah sebagai berikut:

- Menampilkan informasi *user* yang terdapat di *server PVE* menggunakan sintak penulisan:

```
# pvesh get /access/users
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
root@pve:~# pvesh get /access/users
```

userid	comment	email	enable	expire	firstname	keys	lastname
ali@pve	ali		1	0	ali		
hasan@pve	hasan		1	0	hasan		
root@pam		putu.hariyadi@gmail.com	1	0			

Output menunjukkan bahwa *server PVE* memiliki 3 (tiga) *user* yaitu dengan **userid** **ali** dan **hasan** berjenis **PVE Authentication Server** serta **root** berjenis **Pluggable Authentication Module (PAM)**.

2. Membuat *user* baru di *server PVE* dengan sintak penulisan:

```
# pvesh create /access/users
```

Selain itu diperlukan pengaturan beberapa parameter antara lain:

- a) `userid` digunakan untuk mengatur ID dari *user* yang digunakan sebagai nama *login* pengguna dan jenis otentikasi atau *realm* yang digunakan berupa **pve** atau **pam**, sebagai contoh **badu@pve**.
- b) `password` digunakan untuk mengatur sandi login dari *user*, sebagai contoh **rahasia**.
- c) `enable` digunakan untuk mengaktifkan atau menonaktifkan akun *user* berupa nilai *boolean* 1 untuk aktif (*enable*) dan 0 untuk nonaktif (*disable*). Secara default bernilai 1 (aktif).
- d) `firstname` digunakan untuk mengatur nama depan *user*, sebagai contoh **Badu**.
- e) `lastname` digunakan untuk mengatur nama belakang *user*, sebagai contoh **Bumigora**.
- f) `comment` digunakan untuk memberikan deskripsi singkat terkait *user*, sebagai contoh **Angkatan 2019**.

sehingga menjadi:

```
# pvesh create /access/users \
-userid badu@pve -password rahasia \
-firstname Badu -lastname Bumigora \
-comment "Angkatan 2019"
```

Hasil eksekusi dari perintah tersebut pada *terminal*, seperti terlihat pada gambar berikut:

```
root@pve:~# pvesh create /access/users \
> -userid badu@pve -password rahasia \
> -firstname Badu -lastname Bumigora \
> -comment "Angkatan 2019"
```

- Memverifikasi hasil dari pembuatan *user* baru yang memiliki ID **badu@pve** dengan mengeksekusi perintah berikut:

```
# pvesh get /access/users
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
root@pve:~# pvesh get /access/users
```

userid	comment	email	enable	expire	firstname	keys	lastname
ali@pve	ali		1	0	ali		
badu@pve	Angkatan 2019		1	0	Badu		Bumigora
hasan@pve	hasan		1	0	hasan		
root@pam		putu.hariyadi@gmail.com	1	0			

Output menunjukkan bahwa *user* dengan **userid badu@pve** telah berhasil dibuat.

- Mengubah sandi login dari *user* tertentu di *server* PVE dengan sintak penulisan:

```
# pvesh set /access/password
```

Selain itu diperlukan pengaturan 2 (dua) parameter yaitu:

- userid** digunakan untuk menentukan ID atau nama login dari *user* yang akan diubah sandinya, sebagai contoh **badu@pve**.
- password** digunakan untuk mengatur sandi login baru bagi *user*, sebagai contoh **12345678**.

sehingga menjadi:

```
# pvesh set /access/password \
-userid badu@pve -password 12345678
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
root@pve:~# pvesh set /access/password \
> -userid badu@pve -password 12345678
```

D. Manajemen Permission atau Access Control List (ACL) dan Menghapus User

Adapun langkah-langkah manajemen *permission* atau ACL dari *user* terhadap objek dan menghapus *user* adalah sebagai berikut:

1. Menampilkan informasi *permission* yang terdapat di *server* PVE menggunakan sintak penulisan:

```
# pvesh get /access/acl
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
root@pve:~# pvesh get /access/acl
```

path	roleid	type	ugid	propagate
/vms/100	PVEVMUser	user	ali@pve	1
/vms/100	PVEVMAAdmin	user	hasan@pve	1

Output menunjukkan bahwa *server* PVE memiliki 2 (dua) ACL untuk **userid ali@pve** dan **hasan@pve**.

2. Membuat *permission* bagi *user* tertentu sehingga dapat mengakses objek di *server* PVE dengan sintak penulisan:

```
# pvesh set /access/acl
```

Selain itu diperlukan pengaturan 3 (tiga) parameter antara lain:

- a) *path* digunakan untuk mengatur ijin akses ke objek tertentu, sebagai contoh untuk VM dengan ID **100** yaitu **/vms/100**.
- b) *user* digunakan untuk menentukan *user* yang diberikan akses terhadap *path* **/vms/100**, sebagai contoh **badu@pve**.
- c) *roles* digunakan untuk mengatur *roles* atau daftar hak akses yang dialokasikan untuk *user badu@pve*, sebagai contoh **PVEVMUser**.

sehingga menjadi:

```
# pvesh set /access/acl \
```

```
-path /vms/100 -users badu@pve -roles PVEVMUser
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
root@pve:~# pvesh set /access/acl \
> -path /vms/100 -users badu@pve -roles PVEVMUser
```

- Memverifikasi hasil dari pembuatan *permission* baru bagi user **badu@pve** dengan mengeksekusi perintah berikut:

```
# pvesh get /access/acl
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
root@pve:~# pvesh get /access/acl
```

path	roleid	type	ugid	propagate
/vms/100	PVEVMUser	user	badu@pve	1
/vms/100	PVEVMAdmin	user	hasan@pve	1
/vms/100	PVEVMUser	user	ali@pve	1

Output menunjukkan bahwa *permission* untuk **ugid badu@pve** terhadap objek **VM ID 100** dengan **roleid PVEVMUser** telah berhasil dibuat.

- Menghapus *permission* yang dimiliki oleh *user* terhadap objek tertentu di server PVE dengan sintak penulisan:

```
# pvesh set /access/acl
```

Selain itu diperlukan pengaturan 4 (empat) parameter antara lain:

- `delete` digunakan untuk menentukan operasi penghapusan *permission*.
- `path` digunakan untuk mengatur ijin akses ke objek tertentu yang akan dihapus, sebagai contoh untuk VM dengan ID **100** yaitu **/vms/100**.
- `user` digunakan untuk menentukan *user* yang akan dihapus aksesnya terhadap *path* **/vms/100**, sebagai contoh **badu@pve**.
- `roles` digunakan untuk mengatur *roles* atau daftar hak akses yang dihapus alokasinya dari *user badu@pve*, sebagai contoh **PVEVMUser**.

sehingga menjadi:

```
# pvesh set /access/acl \
-delete -path /vms/100 -users badu@pve -roles PVEVMUser
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
root@pve:~# pvesh set /access/acl \
> -delete -path /vms/100 -users badu@pve -roles PVEVMUser
```

- Memverifikasi hasil dari penghapusan *permission* dengan mengeksekusi perintah berikut:

```
# pvesh get /access/acl
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
root@pve:~# pvesh get /access/acl
```

path	roleid	type	ugid	propagate
/vms/100	PVEVMUser	user	ali@pve	1
/vms/100	PVEVMAdmin	user	hasan@pve	1

Output menunjukkan bahwa tidak terdapat *permission* dari user ID **badu@pve** terhadap objek **VM ID 100** dengan **roles PVEVMUser** sehingga *permission* tersebut telah berhasil dihapus.

- Menghapus *user* tertentu di *server* PVE dengan sintak penulisan:

```
# pvesh delete /access/users/{userid}
```

Ubah {userid} dengan ID atau nama login dari *user* yang akan dihapus, sebagai contoh **badu@pve**, sehingga menjadi:

```
# pvesh delete /access/users/badu@pve
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
root@pve:~# pvesh delete /access/users/badu@pve
```

- Memverifikasi hasil dari penghapusan *user* yang memiliki ID **badu@pve** dengan mengeksekusi perintah berikut:

```
# pvesh get /access/users
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
root@pve:~# pvesh get /access/users
```

userid	comment	email	enable	expire	firstname	keys	lastname
ali@pve	ali		1	0	ali		
hasan@pve	hasan		1	0	hasan		
root@pam		putu.hariyadi@gmail.com	1	0			

Output menunjukkan bahwa tidak terdapat *user* dengan **userid badu@pve** sehingga *user* tersebut telah berhasil dihapus.

BAB VIII

PENGENALAN ANSIBLE DAN ANSIBLE PROXMOX MODULE

Bab ini terdiri dari 4 (empat) bagian yaitu pengenalan *ansible*, terminologi *ansible*, arsitektur *ansible*, dan *ansible proxmox module*.

A. PENGENALAN ANSIBLE

Menurut situs [Ansible](#), **Ansible** merupakan mesin otomatisasi Teknologi Informasi (TI) sederhana yang dapat mengotomatisasi *cloud provisioning*, manajemen konfigurasi, penerapan aplikasi, *intra-service orchestration* dan kebutuhan TI lainnya.

Menurut situs [Edureka](#), keuntungan menggunakan *Ansible* antara lain:

1. **Simple**, menggunakan sintak penulisan yang ditulis menggunakan YAML yang disebut dengan *playbook*. YAML merupakan singkatan dari “**YAML Ain’t markup language**”.
2. **Agentless**, tidak diperlukan *agent* atau perangkat lunak khusus untuk diinstalasi pada host yang di otomasi.
3. **Powerful & Flexible**, memiliki banyak modul untuk manajemen infrastruktur, jaringan, sistem operasi dan layanan.
4. **Efficient**, tidak ada perangkat lunak yang diperlukan untuk instalasi pada *server* sehingga lebih banyak sumber daya yang dapat digunakan oleh aplikasi.

Ansible dapat melakukan hal-hal berikut:

1. **Provisioning**

Ansible memastikan paket-paket yang dibutuhkan akan diunduh dan diinstalasi sehingga aplikasi dapat diterapkan.

2. **Configuration Management**

Menetapkan dan mempertahankan konsistensi dari kinerja produk dengan mencatat dan memperbaharui informasi lengkap yang menjelaskan perangkat keras dan lunak yang digunakan oleh perusahaan. Seperti versi dan pembaharuan yang telah diterapkan pada

paket perangkat lunak yang terinstalasi dan lokasi serta alamat jaringan dari perangkat keras.

3. **Application Deployment**

Ansible dapat memajemen secara efektif keseluruhan *life cycle* dari aplikasi, mulai dari *development* sampai produksi.

4. **Security and Compliance**

Kebijakan keamanan dapat didefinisikan pada *Ansible* sehingga proses pemindaian dan pemulihan kebijakan ke lokasi dapat diintegrasikan ke dalam proses secara otomatis.

5. **Orchestration**

Ansible menyediakan *orchestration* dalam arti menyelaraskan permintaan bisnis dengan aplikasi, data, dan infrastruktur. Ini mendefinisikan kebijakan dan tingkat layanan melalui alur kerja otomatis, penyediaan, dan manajemen perubahan sehingga menciptakan aplikasi selaras dengan infrastruktur yang dapat ditingkatkan dari atas ke bawah berdasarkan kebutuhan setiap aplikasi.

B. TERMINOLOGI ANSIBLE

Menurut situs Edureka, terdapat beberapa terminologi pada *Ansible* yaitu antara lain:

1. **Controller Machine**

Mesin dimana ansible diinstalasi dan bertanggungjawab untuk menjalankan *provisioning* pada server yang dikelola.

2. **Inventory**

File inialisasi yang memuat informasi tentang *server* yang dikelola.

3. **Playbook**

Titik masuk untuk *Ansible provisioning*, dimana otomasi didefinisikan melalui tugas (*tasks*) menggunakan format *YAML*.

4. **Task**

Blok yang mendefinisikan satu prosedur untuk dieksekusi, sebagai contoh instalasi package tertentu.

5. **Module**

Modul merupakan abstraksi dari tugas sistem, seperti berkaitan dengan *package* atau membuat dan mengubah *file*. *Ansible* memiliki banyak modul built-in, namun dapat juga dibuat modul *custom*.

6. **Role**

Cara yang telah ditentukan sebelumnya untuk mengatur *playbook* dan *file* lainnya guna memfasilitasi berbagi pakai dan menggunakan kembali bagian dari *provisioning*.

7. **Play**

Provisioning yang dieksekusi mulai dari awal sampai akhir disebut dengan *play*. Dengan kata lain, eksekusi dari *playbook* disebut dengan *play*.

8. **Facts**

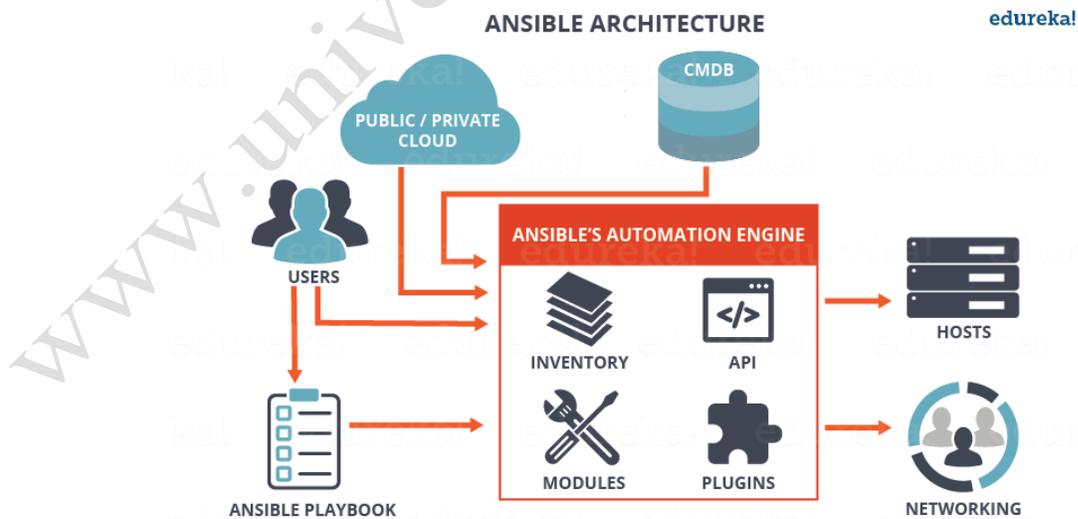
Variable global yang memuat informasi tentang sistem, seperti *interface* jaringan atau sistem operasi.

9. **Handlers**

Digunakan untuk memicu perubahan status dari *service*, seperti *me-restart* atau menghentikan *service*.

C. ARSITEKTUR ANSIBLE

Ansible memiliki arsitektur, seperti terlihat pada gambar berikut:



[Gambar bersumber dari situs edureka](#)

Ansible Automation Engine (AAE) terdiri dari *Inventory*, *Application Programming Interface (API)*, *Modules* dan *Plugins*. Pengguna yang menulis *ansible playbook* berinteraksi secara langsung dengan AAE ketika *playbook* tersebut dieksekusi. *Playbook* ditulis menggunakan format *YAML* yang didalamnya memuat *task* yang akan dieksekusi oleh *Ansible*. *Inventory* merupakan daftar dari *host* atau *node* yang memiliki alamat IP sebagai mesin yang dimanajemen oleh *ansible* dan lokasi eksekusi dari *playbook*. API pada *ansible* digunakan metode *transport* untuk komunikasi dengan layanan *Cloud* baik publik maupun privat. *Module* dieksekusi secara langsung pada mesin yang dimanajemen atau sistem *remote* oleh *ansible* melalui *playbook*. *Module* dapat mengatur *service*, *package*, sumber daya, *file* atau mengeksekusi perintah sistem. Sebaliknya *plugin* dieksekusi pada *control node* atau mesin dimana *ansible* terinstalasi yaitu di dalam proses `/usr/bin/ansible`. *Plugin* menawarkan perluasan dari fitur inti *ansible* seperti transformasi data, *logging output*, koneksi ke *inventory* dan lain-lain. Selain itu *ansible* juga dapat digunakan untuk mengotomasi jaringan (*networking*).

D. ANSIBLE PROXMOX MODULE

Ansible memiliki 3 (tiga) *module* untuk memanajemen objek tertentu pada *Proxmox VE* meliputi:

1. **proxmox**

Digunakan untuk memanajemen *instance* atau *container* di *Proxmox VE cluster* meliputi *create/delete/stop instance*. Kebutuhan untuk dapat menggunakan *module* ini adalah *proxmoxer*, *python* ≥ 2.7 dan *requests* pada *host* yang mengeksekusi *module* tersebut. Parameter-parameter yang dimiliki oleh *module* ini termasuk contoh penggunaannya dapat diakses pada alamat berikut:

https://docs.ansible.com/ansible/latest/modules/proxmox_module.html?highlight=proxmox

2. **proxmox_kvm**

Digunakan untuk manajemen *Qemu (Kernel-based Virtual Machine)* pada *Proxmox VE Cluster* meliputi *create/delete/stop Qemu (KVM) virtual machine*. Kebutuhan untuk dapat menggunakan *module* ini adalah *proxmoxer* dan *requests* pada *host* yang mengeksekusi *module* tersebut. Parameter-parameter yang dimiliki oleh *module* ini termasuk contoh penggunaannya dapat diakses pada alamat berikut:

https://docs.ansible.com/ansible/latest/modules/proxmox_kvm_module.html?highlight=proxmox

3. proxmox_template

Digunakan untuk manajemen *template* dari sistem operasi pada *Proxmox VE cluster*. Kebutuhan untuk dapat menggunakan *module* ini adalah *proxmoxer* dan *requests* pada *host* yang mengeksekusi *module* tersebut. Parameter-parameter yang dimiliki oleh *module* ini termasuk contoh penggunaannya dapat diakses pada alamat berikut:

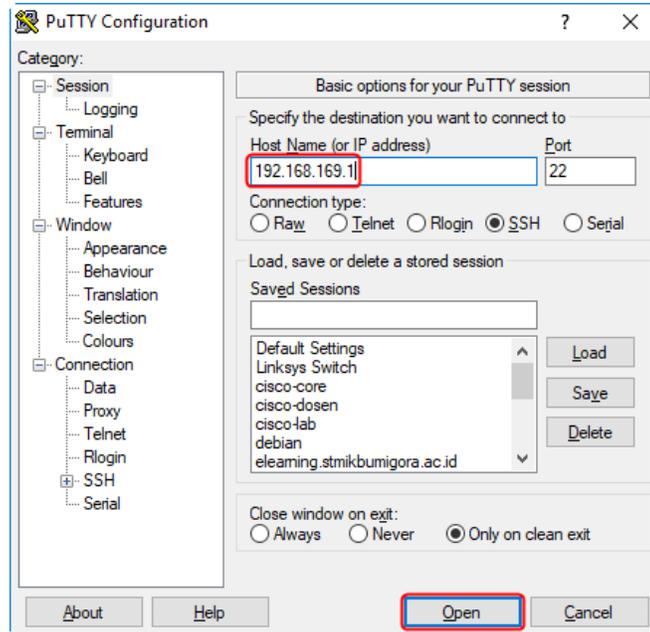
https://docs.ansible.com/ansible/latest/modules/proxmox_template_module.html?highlight=proxmox

Selain tiga *module* tersebut, pada *Ansible* juga terdapat *module command* yang dapat digunakan untuk mengeksekusi perintah-perintah pada target *host* termasuk utilitas **pvesh** dari *Proxmox VE*. *Module* ini dapat memfasilitasi manajemen objek pada *Proxmox VE* yang diotomasi menggunakan *ansible* seperti *user*, *permission* dan lain sebagainya. Parameter-parameter yang dimiliki oleh *module* ini termasuk contoh penggunaannya dapat diakses pada alamat berikut:

https://docs.ansible.com/ansible/latest/modules/command_module.html#command-module

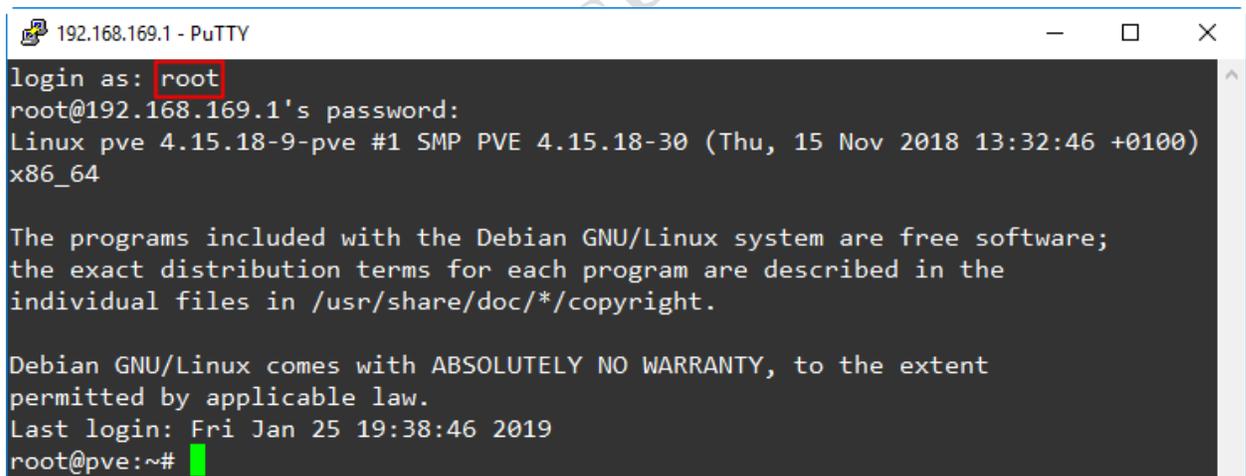
Menindaklanjuti persyaratan kebutuhan perangkat lunak yang harus terinstalasi pada *server PVE* sebelum dapat menggunakan *ansible proxmox module* maka akan dilakukan instalasi paket aplikasi **python-pip** dan **proxmoxer**. Adapun langkah-langkah menginstalasi paket aplikasi **python-pip** dan **proxmoxer** tersebut di *server PVE* adalah sebagai berikut:

1. Jalankan aplikasi *SSH Client*, sebagai contoh menggunakan *Putty*. Tampil kotak dialog *Putty Configuration*. Pada isian **Host Name (or IP Address)**, masukkan alamat IP dari *Server Proxmox* yaitu **192.168.169.1**, seperti terlihat pada gambar berikut:



Klik tombol **Open**.

2. Tampil kotak dialog *PuTTY* yang meminta pengguna untuk melakukan proses otentikasi login ke *Server Proxmox*, seperti terlihat pada gambar berikut:



Pada inputan **login as:**, masukkan “**root**” dan tekan tombol **Enter**. Selanjutnya tampil inputan **password:**, masukkan “**12345678**” dan tekan tombol **Enter**. Apabila proses otentikasi berhasil dilakukan maka akan tampil *shell prompt #*.

3. Menginstalasi paket aplikasi **python-pip** dengan mengeksekusi perintah:

```
# apt -y install python-pip
```

Cuplikan hasil dari proses instalasi *python-pip*, seperti terlihat pada gambar berikut:

```
Setting up gcc-6 (6.3.0-18+deb9ul) ...
Setting up g++-6 (6.3.0-18+deb9ul) ...
Setting up libalgorithm-merge-perl (0.08-3) ...
Setting up python-gi (3.22.0-2) ...
Setting up dpkg-dev (1.18.25) ...
Setting up libalgorithm-diff-xs-perl (0.04-4+b2) ...
Setting up python-all-dev (2.7.13-2) ...
Setting up python-cryptography (1.7.1-3+deb9ul) ...
Setting up python-secretstorage (2.3.1-2) ...
Setting up python-keyring (10.1-1) ...
Setting up gcc (4:6.3.0-4) ...
Setting up g++ (4:6.3.0-4) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Setting up build-essential (12.3) ...
Processing triggers for libc-bin (2.24-11+deb9ul) ...
root@pve:~#
```

4. Menginstalasi paket aplikasi **proxmoxer** menggunakan **pip** dengan mengeksekusi perintah:

```
# pip install proxmoxer
```

Cuplikan hasil dari proses instalasi *proxmoxer*, seperti terlihat pada gambar berikut:

```
root@pve:~# pip install proxmoxer
Collecting proxmoxer
  Downloading https://files.pythonhosted.org/packages/33/92/0319a36acc8eab45fe0e
d20bb19b567ed8fa50b635f5ca2b2098fd531312/proxmoxer-1.0.3.tar.gz
Building wheels for collected packages: proxmoxer
  Running setup.py bdist_wheel for proxmoxer ... done
  Stored in directory: /root/.cache/pip/wheels/e7/b9/b9/4c703be742f1bdcfe62cb763
7c995b051d9d53bf324fe62a32
Successfully built proxmoxer
Installing collected packages: proxmoxer
Successfully installed proxmoxer-1.0.3
```

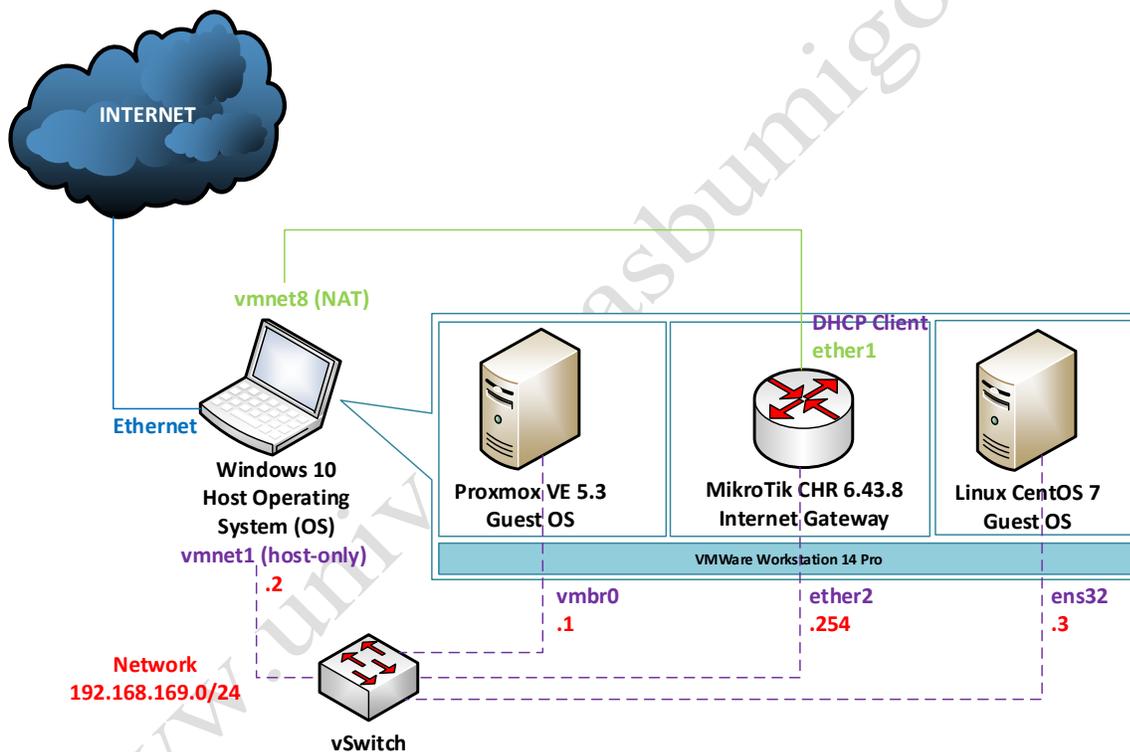
5. Keluar dari SSH dengan mengeksekusi perintah:

```
# exit
```

BAB IX

INSTALASI DAN KONFIGURASI LINUX CENTOS 7 SEBAGAI ANSIBLE CONTROL MACHINE PADA VMWARE WORKSTATION 14

Bab ini terdiri dari 2 (dua) bagian yaitu (a) Instalasi dan konfigurasi *Linux CentOS 7* pada *VMWare Workstation 14*, (b) Instalasi dan konfigurasi *Ansible* pada *CentOS 7* sebagai *control machine*. Rancangan jaringan ujicoba yang digunakan untuk menerapkan *Ansible Control Machine*, seperti terlihat pada gambar berikut:



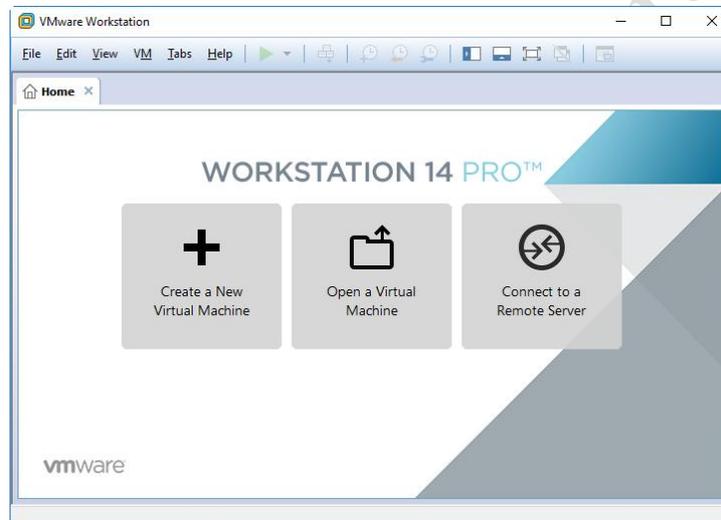
Pada *VMWare Workstation 14* akan dibuat *guest virtual machine* dengan sistem operasi *Linux CentOS 7*. Alamat IP yang digunakan pada *interface* jaringan **ens32** dari *Linux CentOS 7* adalah **192.168.169.3/24** dengan **default gateway** dan **server DNS 192.168.169.254** sehingga dapat terkoneksi ke *Internet* melalui *router MikroTik CHR Internet Gateway*. Koneksi *Internet* diperlukan

ketika menginstalasi paket aplikasi *editor nano* dan *openssh* agar Linux CentOS 7 dapat di *remote access* dari *windows client* serta *ansible*.

A. INSTALASI & KONFIGURASI LINUX CENTOS 7 PADA VMWARE WORKSTATION 14

Adapun langkah-langkah instalasi *CentOS 7* pada *VMWare Workstation 14* adalah sebagai berikut:

1. Jalankan aplikasi *VMWare Workstation 14* melalui **Start > VMWare > VMWare Workstation Pro**.
2. Tampil aplikasi *VMWare Workstation* seperti terlihat pada gambar berikut:



3. Untuk membuat *virtual machine* baru pilih menu **File > New Virtual Machine ...**, seperti terlihat pada gambar berikut:

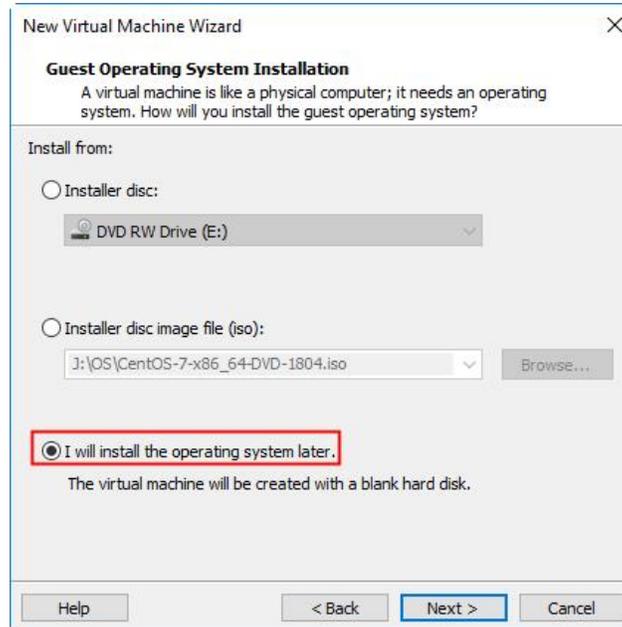


4. Tampil kotak dialog *New Virtual Machine Wizard* untuk menentukan jenis konfigurasi *virtual machine* yang ingin dibuat, seperti terlihat pada gambar berikut:



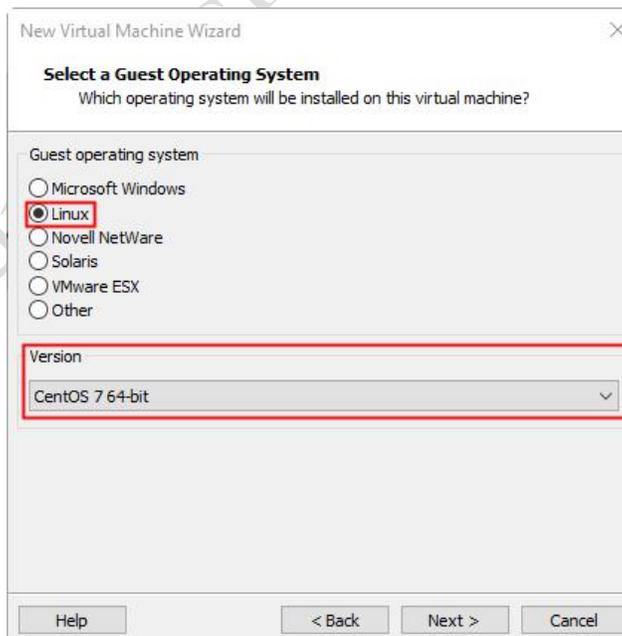
Terdapat 2 pilihan jenis konfigurasi yang dapat dipilih yaitu *Typical (recommended)* dan *Custom (advanced)*. Jenis konfigurasi *Typical* disarankan untuk dipilih ketika ingin membuat virtual machine melalui beberapa tahapan dengan mudah. Sebaliknya jenis konfigurasi *Custom* akan memberikan pilihan pengaturan lanjutan seperti penentuan jenis *controller SCSI*, jenis *virtual disk* dan kompatibilitas dengan produk *VMWare* versi sebelumnya. Pilih **Typical**, dan klik tombol **Next >** untuk melanjutkan.

5. Tampil kotak dialog *Guest Operating System Installation* untuk menentukan bagaimana cara instalasi sistem operasi dilakukan. Terdapat 3 pilihan yaitu *Install from: Installer disc* untuk menginstalasi dari media disc seperti CD/DVD, *Install from: Installer disc image file (iso)* untuk menginstalasi dari file ISO, dan *I will install the operating system later* untuk mempersiapkan virtual machine dengan hardisk kosong tanpa melakukan instalasi sistem operasi, seperti terlihat pada gambar berikut:



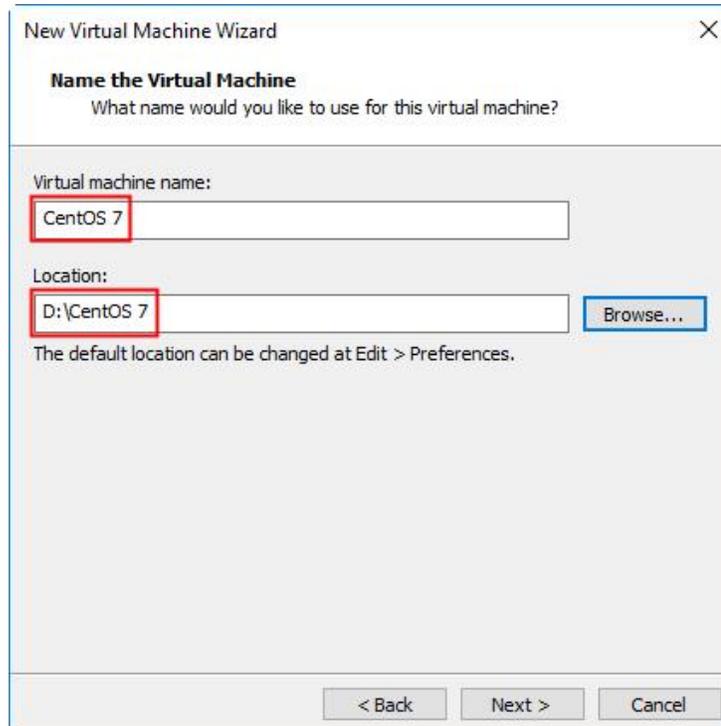
Pilih *I will install the operating system later*, dan klik tombol **Next >** untuk melanjutkan.

6. Tampil kotak dialog *Select a Guest Operating System* untuk menentukan jenis sistem operasi yang akan diinstalasi pada virtual machine yang dibuat, seperti ditunjukkan pada gambar berikut:



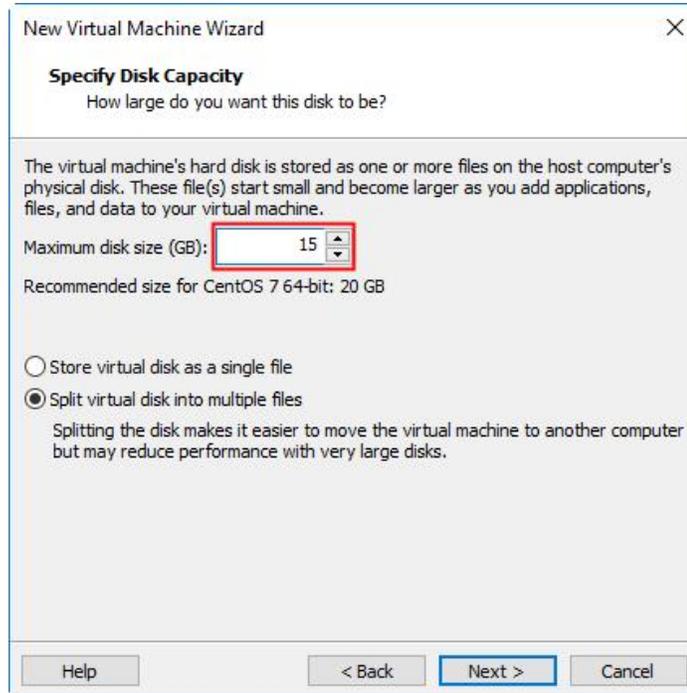
Pilih *Linux* pada bagian *Guest operating system*, dan *CentOS 7 64 bit* pada bagian *Version*.
Klik tombol **Next >** untuk melanjutkan.

7. Tampil kotak dialog *Name the Virtual Machine* untuk menentukan nama pengenal *virtual machine* dan menentukan lokasi penyimpanan file *virtual machine* yang dibuat, seperti terlihat pada gambar berikut:



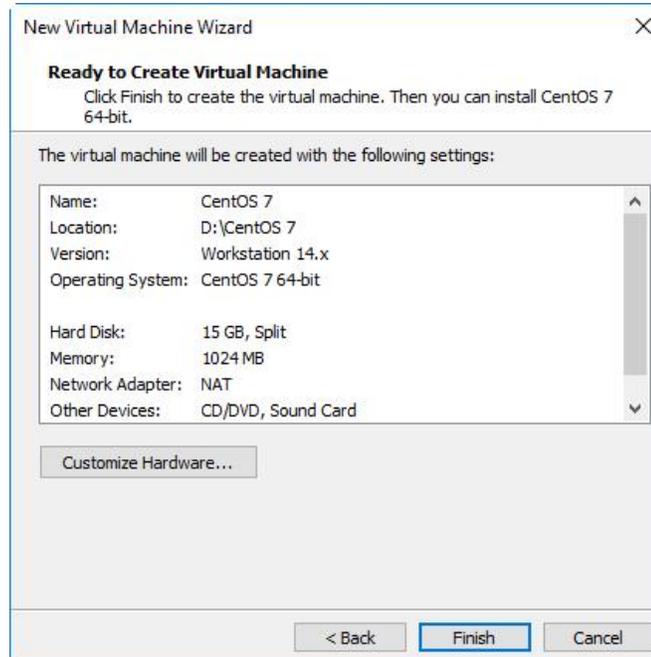
Pada bagian *Virtual machine name* masukkan nama pengenal virtual machine, sebagai contoh **CentOS 7**. Sedangkan pada bagian *Location* tentukan lokasi penyimpanan file virtual machine yang dibuat dengan cara menekan tombol *Browse ...* sebagai contoh diletakkan di **D:\CentOS 7**. Klik tombol **Next >** untuk melanjutkan.

8. Tampil kotak dialog *Specify Disk Capacity* untuk menentukan kapasitas media penyimpanan yang dialokasikan untuk virtual machine yang dibuat, seperti terlihat pada gambar berikut:



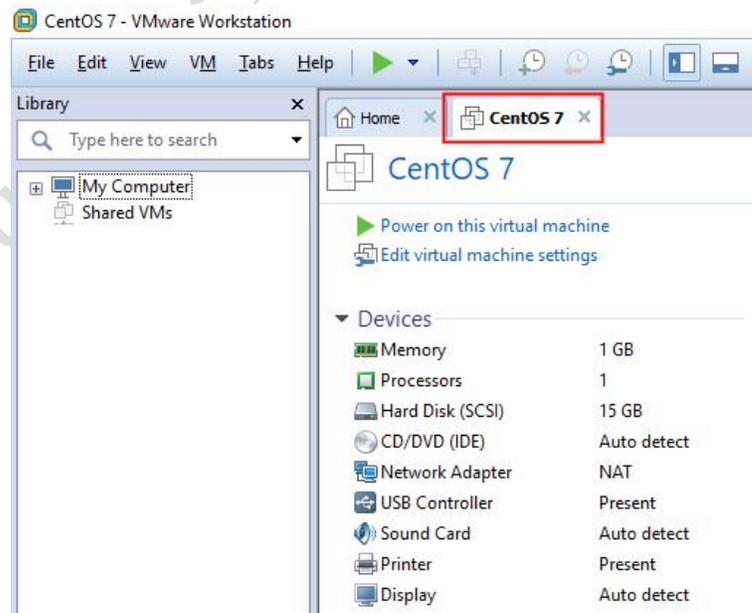
Pada bagian Maximum disk size (GB) masukkan kapasitas media penyimpanan (hardisk) yang dialokasikan untuk virtual machine yang dibuat, sebagai contoh 15 GB. Klik tombol **Next >** untuk melanjutkan.

9. Tampil kotak dialog *Ready to Create Virtual Machine* yang menampilkan informasi ringkasan pengaturan yang telah ditentukan untuk virtual machine yang akan dibuat, seperti terlihat pada gambar berikut:

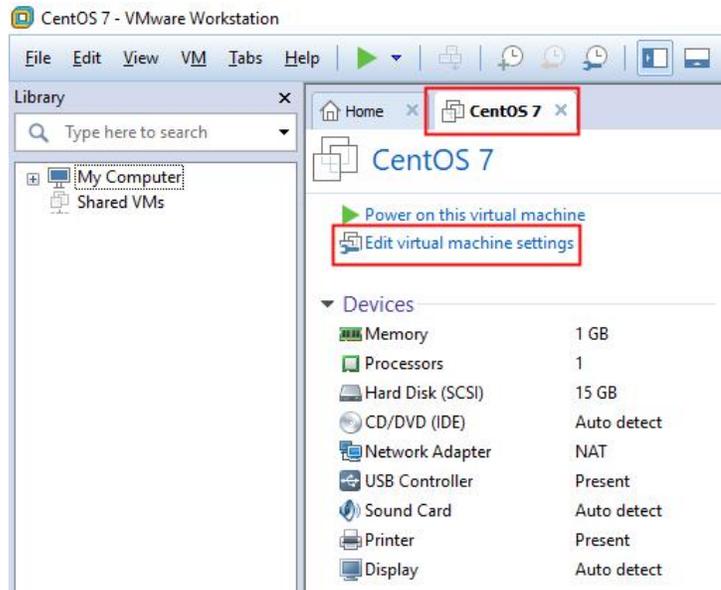


Klik tombol **Finish** untuk membuat virtual machine. Selanjutnya Anda dapat menginstalasi sistem operasi Linux CentOS 7.

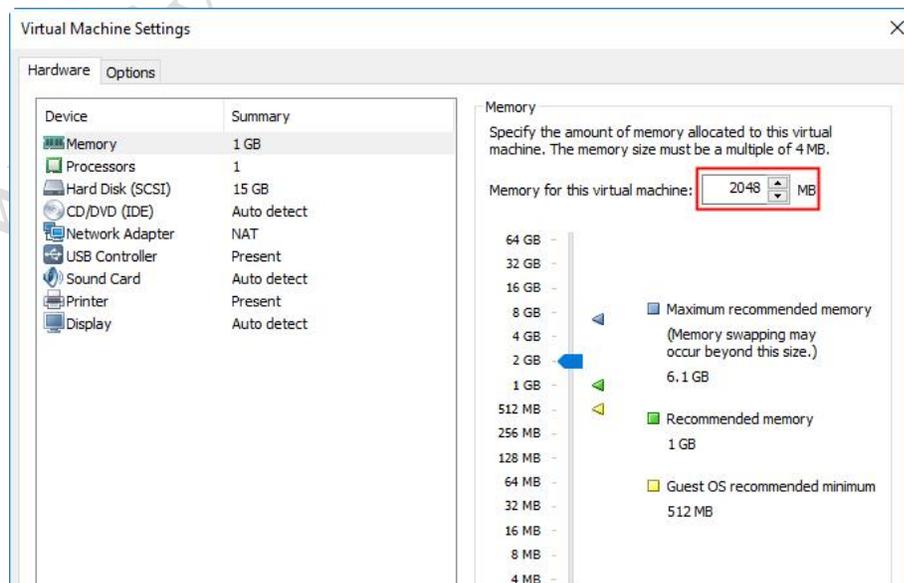
10. Tampil kotak dialog yang menampilkan *virtual machine* yang telah berhasil dibuat yaitu dengan nama pengenal *CentOS 7*, seperti terlihat pada gambar berikut:



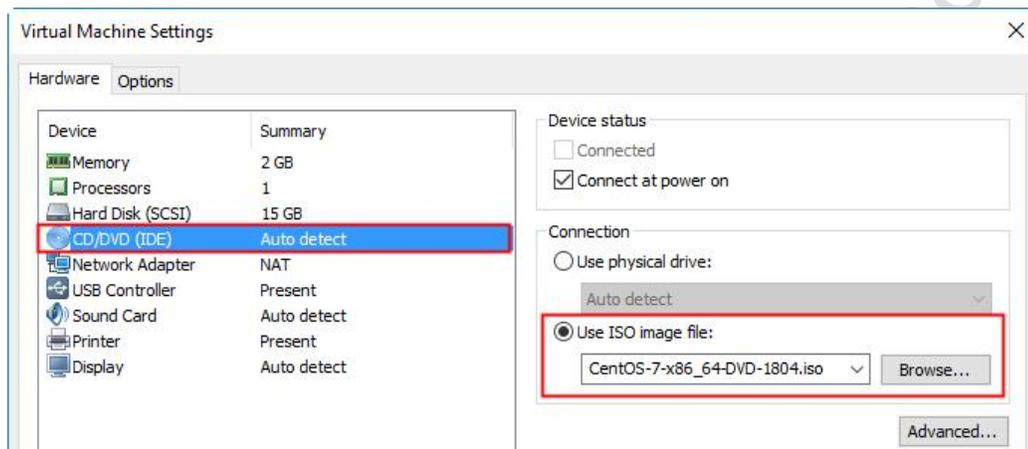
Selanjutnya klik *Edit virtual machine settings* untuk melakukan perubahan pada pengaturan *virtual machine* untuk beberapa komponen hardware, seperti terlihat pada gambar berikut:



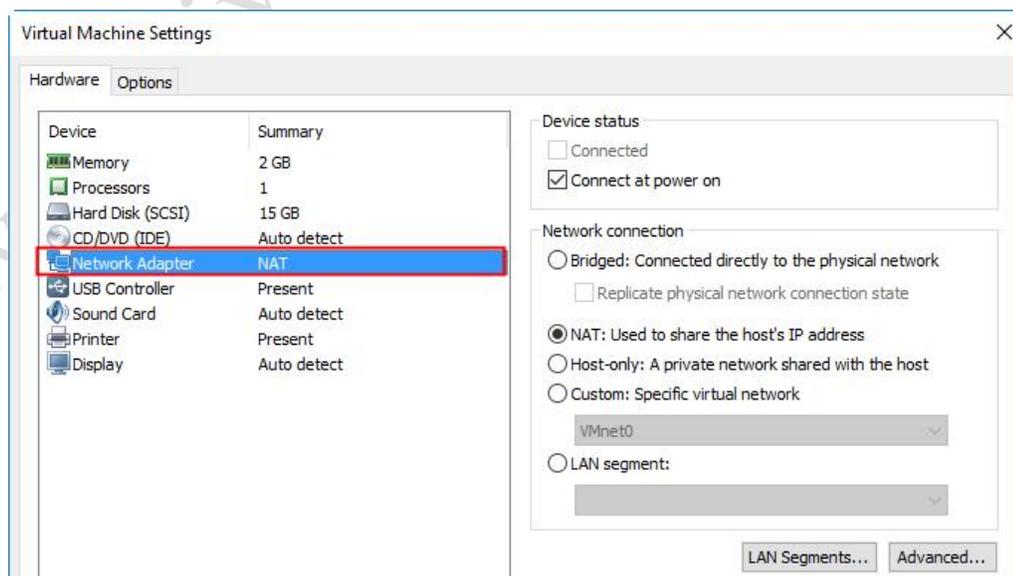
11. Tampil kotak dialog *Virtual Machine Settings*. Pada tab *Hardware* di panel sebelah kiri pilih *Memory*. Selanjutnya pada panel detail sebelah kanan lakukan penyesuaian ukuran memori yang dialokasikan untuk *virtual machine* di parameter *Memory for this virtual Machine* sebagai contoh dialokasikan 2048 MB atau 2 GB, seperti terlihat pada gambar berikut:



12. Pada tab *Hardware* di panel sebelah kiri dari *Virtual Machine Settings* pilih *CD/DVD (IDE)* untuk mengarahkan ke lokasi penyimpanan file ISO dari Linux CentOS 7. Selanjutnya pada panel sebelah kanan akan muncul detail pengaturan CD/DVD. Pada bagian *Connection*, pilih *Use ISO image file*, dan klik tombol *Browse...* untuk mengarahkan ke lokasi penyimpanan file ISO dari Linux CentOS 7 DVD yang akan digunakan sebagai media sumber instalasi, sebagai contoh terdapat di drive **D:\CentOS 7\CentOS-7-x86_64-DVD-1804.iso**, seperti terlihat pada gambar berikut:

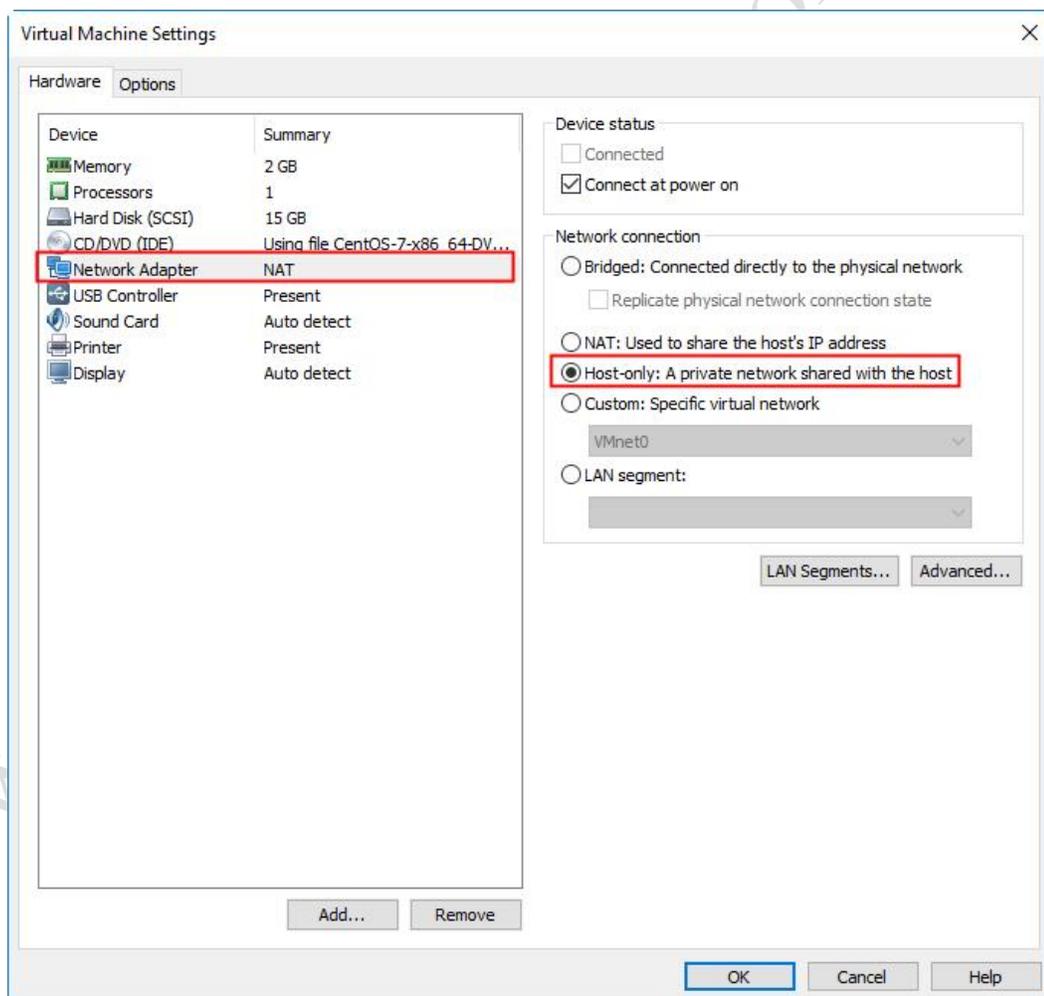


- Pada tab *Hardware* di panel sebelah kiri dari *Virtual Machine Settings* pilih *Network Adapter*. Selanjutnya pada panel sebelah kanan akan muncul detail pengaturan *Network Adapter*, seperti terlihat pada gambar berikut:



Pada bagian *Network connection* beberapa pilihan jenis koneksi jaringan yang dapat digunakan oleh *Network Adapter* yaitu *Bridged* (untuk dapat terhubung secara langsung ke jaringan fisik), *Network Address Translation (NAT)* (untuk berbagi pakai alamat IP dari host), *Host-only* (untuk terhubung ke jaringan privat yang dibagi pakai dengan host), dan *Custom* (untuk secara spesifik menentukan virtual network yang ingin digunakan). Secara default telah terpilih *Network Connection* dengan jenis **NAT** untuk *Network Adapter* dari *VM CentOS 7*.

Lakukan perubahan *Network Connection* untuk *Network Adapter* dari *VM CentOS 7* dari NAT menjadi *Host-only*, seperti terlihat pada gambar berikut:

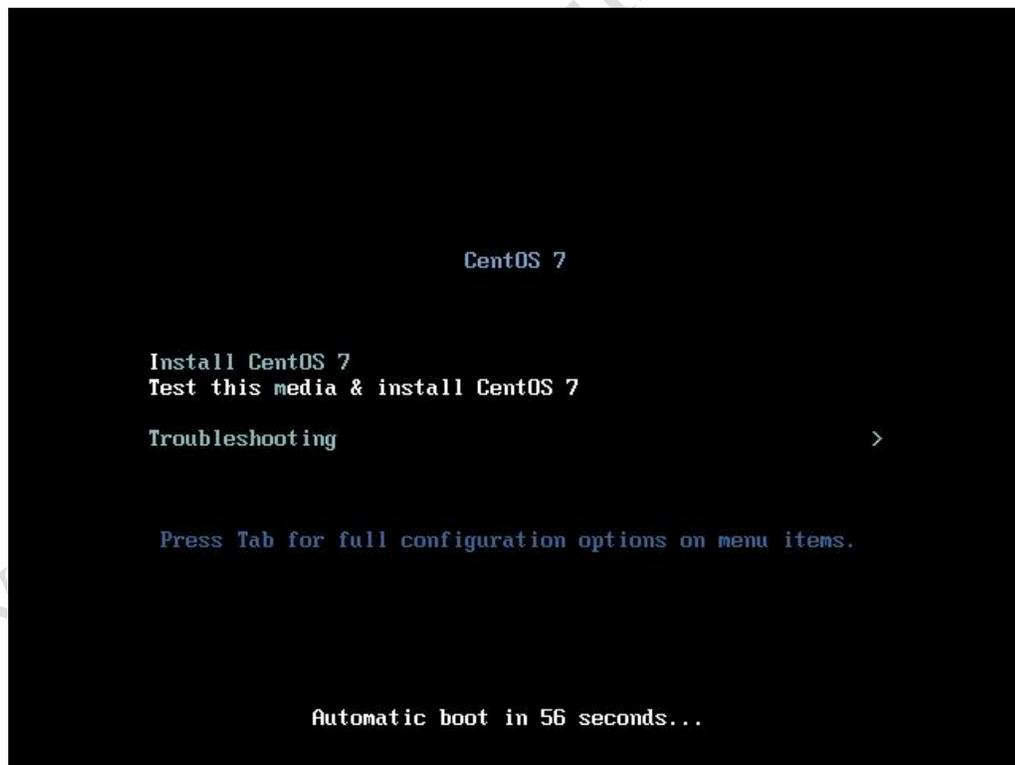


Klik tombol **OK** untuk menutup kotak dialog *Virtual Machine Settings*.

Selanjutnya klik **Power on this virtual machine** untuk menghidupkan virtual machine dan memulai instalasi Linux CentOS 7 pada virtual machine yang telah dibuat, seperti terlihat pada gambar berikut:



13. Tampil menu awal instalasi *CentOS 7* untuk menentukan pilihan instalasi yang akan dilakukan, seperti terlihat pada gambar berikut:



Terdapat beberapa pilihan yang tampil yaitu *Install CentOS 7* dan *Test this media & install CentOS 7* serta *Troubleshooting*. Gunakan tombol panah atas dan bawah untuk navigasi pada pilihan yang ada.

Pilih *Install CentOS 7* dan tekan tombol **Enter** untuk melanjutkan instalasi.

14. Tampil *Welcome To CentOS 7* yang menampilkan pilihan bahasa yang akan digunakan selama proses instalasi, seperti terlihat pada gambar berikut:



Terlihat telah terpilih **English (United States)**. Terdapat pula pilihan **Bahasa Indonesia**. Klik tombol *Continue* untuk melanjutkan instalasi.

15. Tampil **Installation Summary** yang memuat berbagai pilihan pengaturan terkait sistem *CentOS 7* yang terdiri dari 3 bagian yaitu *Localization*, *Software* dan *System*, seperti terlihat pada gambar berikut:



16. Pada layar *Installation Summary* bagian *Localization*, pilih **Date & Time** untuk melakukan pengaturan tanggal dan waktu dari sistem, seperti terlihat pada gambar berikut:



Selanjutnya akan tampil pengaturan zone waktu dapat dilakukan dengan memilih menu *dropdown* atau melalui gambar peta. Untuk Waktu Indonesia Barat (WIB) pilih Asia/Jakarta, untuk Waktu Indonesia Tengah (WITA) pilih Asia/Makassar, sedangkan Wilayah Indonesia Timur (WIT) pilih Asia Jayapura.

Sebagai contoh pilih *Asia/Makassar*, seperti terlihat pada gambar berikut:



Klik tombol **Done** untuk melanjutkan.

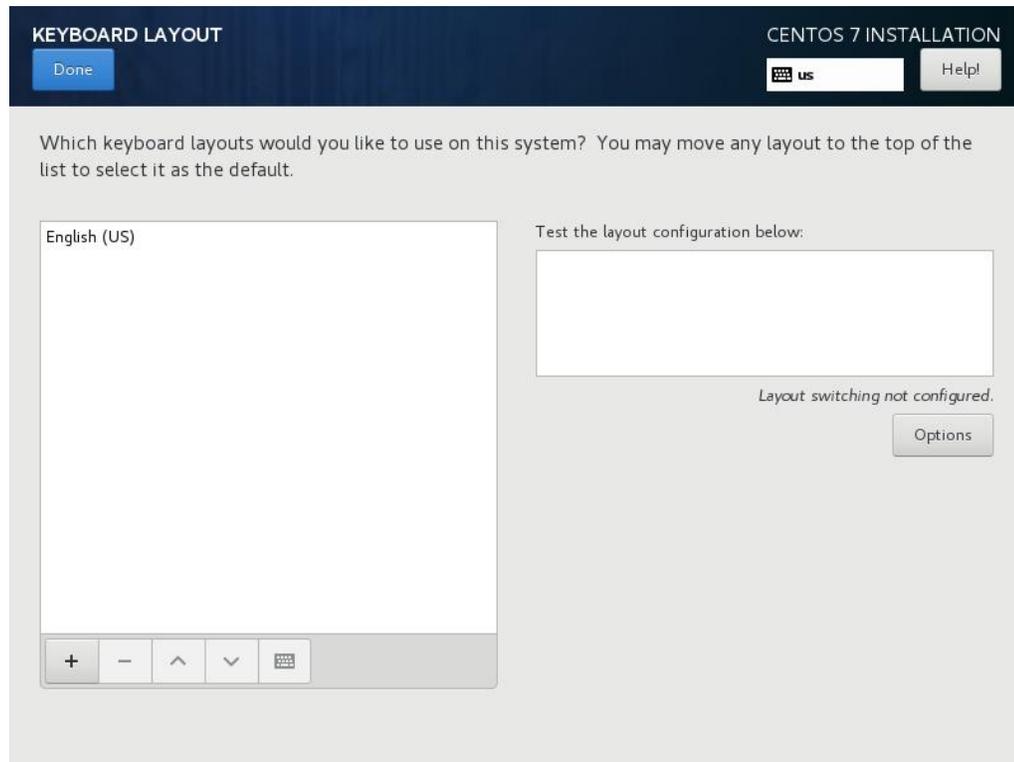
17. Pada layar *Installation Summary* bagian *Localization*, terdapat pilihan **Keyboard** untuk melakukan pengaturan jenis *keyboard* yang digunakan oleh sistem, seperti terlihat pada gambar berikut:



Terlihat saat ini pengaturan jenis *keyboard* menggunakan *English (US)*.

Apabila jenis *keyboard* yang digunakan berbeda maka penyesuaian dapat dilakukan dengan cara klik pada **Keyboard**. Selanjutnya akan tampil tombol navigasi untuk

menambah (+) dan menghapus (-) keyboard layout, serta mengatur urutan *keyboard layout* yang digunakan, seperti terlihat pada gambar berikut:



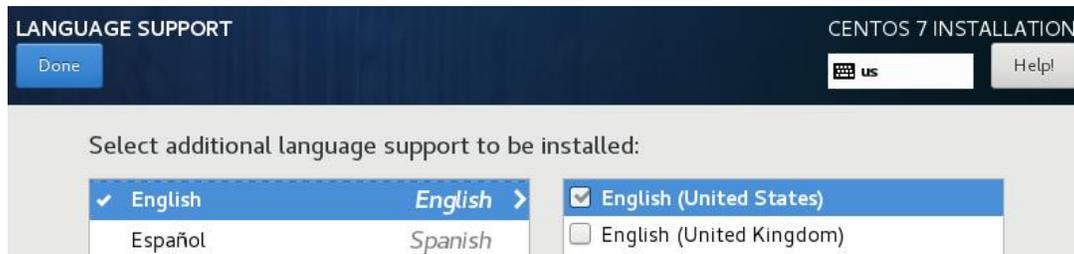
Klik tombol **Done** untuk melanjutkan.

18. Pada layar *Installation Summary* bagian *Localization*, terdapat pilihan **Language Support** untuk melakukan pengaturan dukungan bahasa dari sistem, seperti terlihat pada gambar berikut:

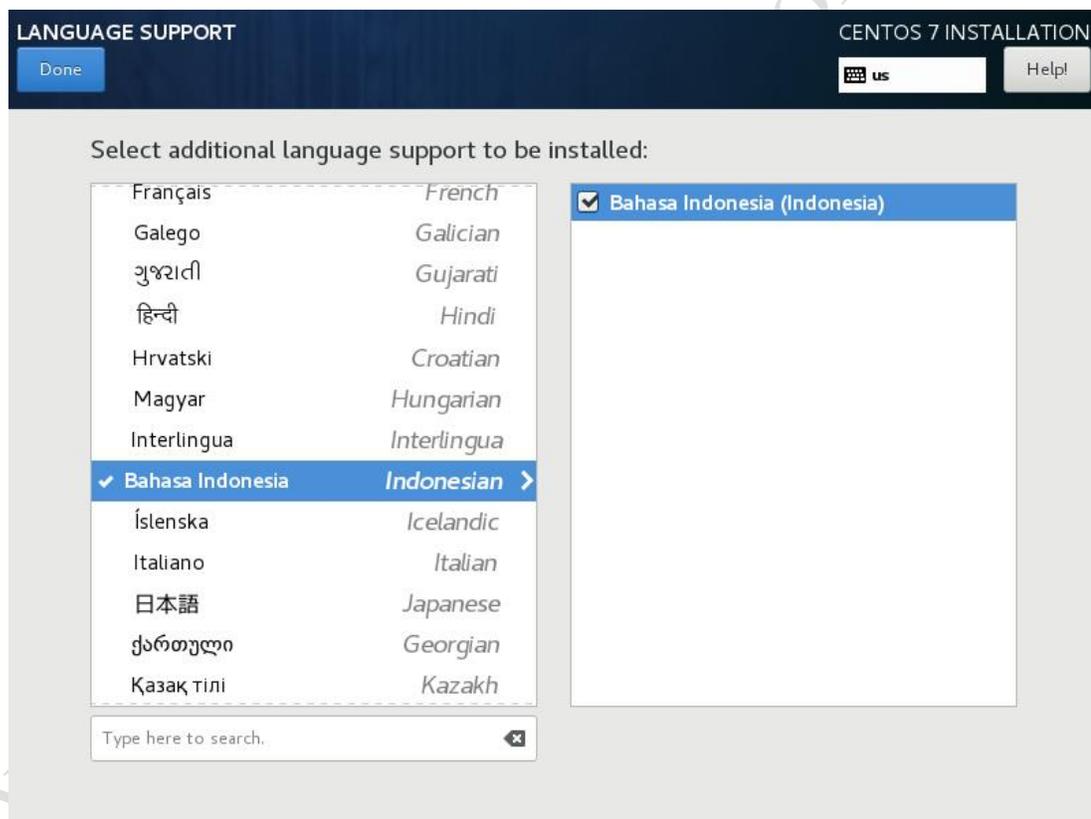


Terlihat saat ini sistem menggunakan bahasa *English (United States)*. Apabila ingin dilakukan penambahan dukungan bahasa yang digunakan maka dapat dilakukan dengan

cara klik pada **Language Support**. Selanjutnya akan tampil pilihan dukungan bahasa yang dapat ditambahkan, seperti terlihat pada gambar berikut:

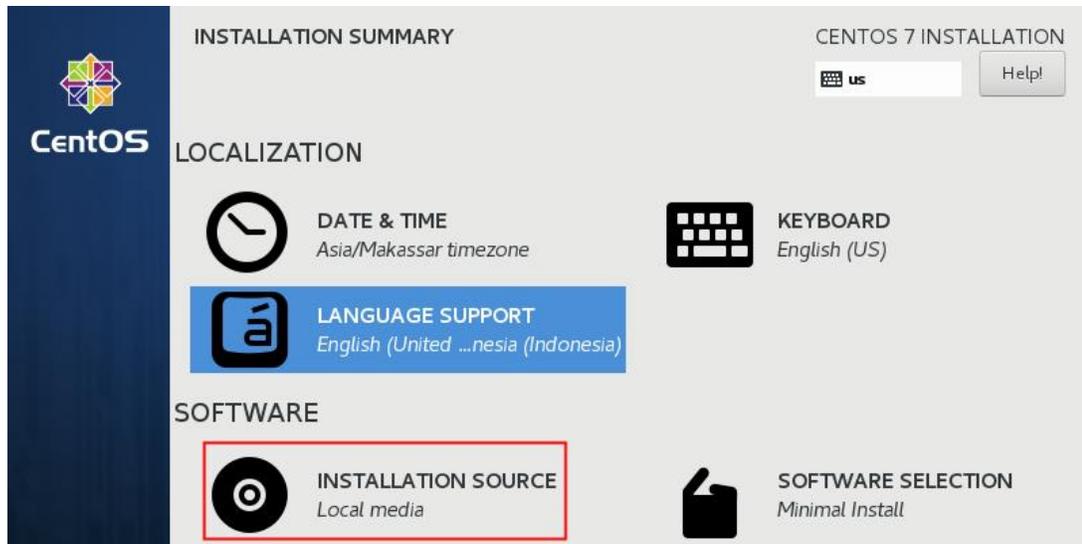


Sebagai contoh tambahkan dukungan untuk Bahasa **Indonesia**, seperti terlihat pada gambar berikut:

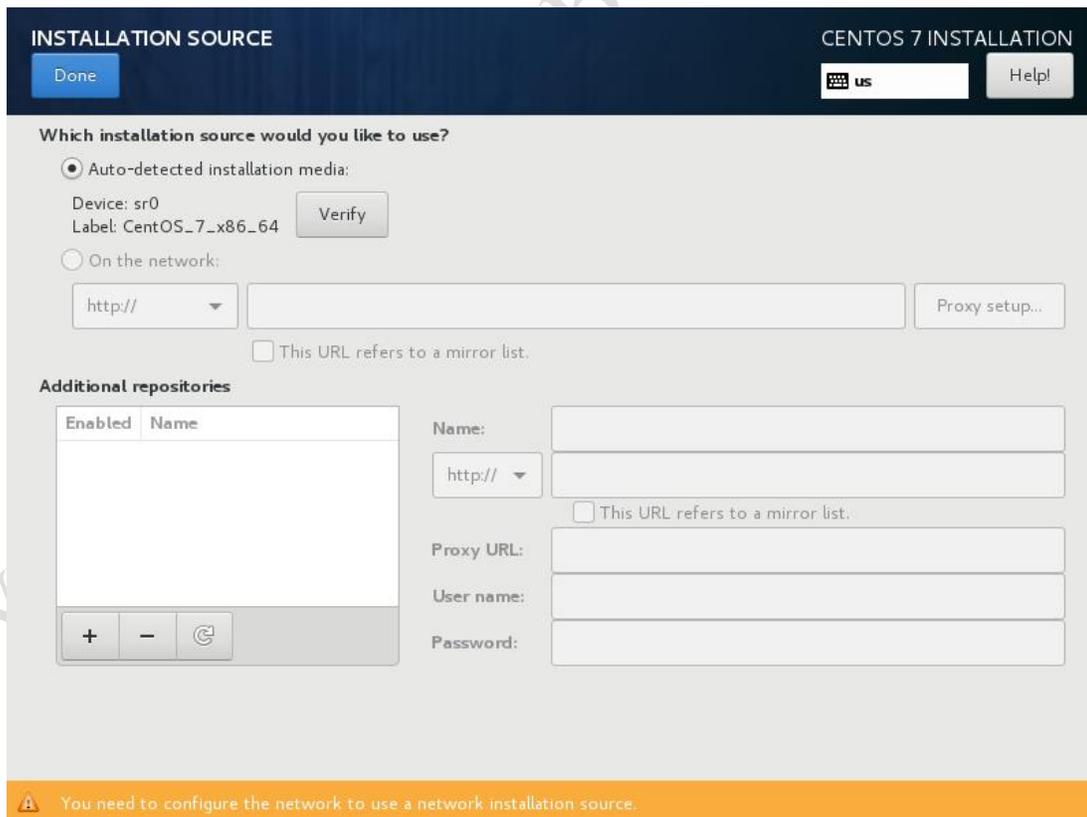


Klik tombol **Done** untuk melanjutkan.

19. Pada layar *Installation Summary* bagian *Software*, terdapat pilihan **Installation Source** untuk mengatur sumber media instalasi, seperti terlihat pada gambar berikut:



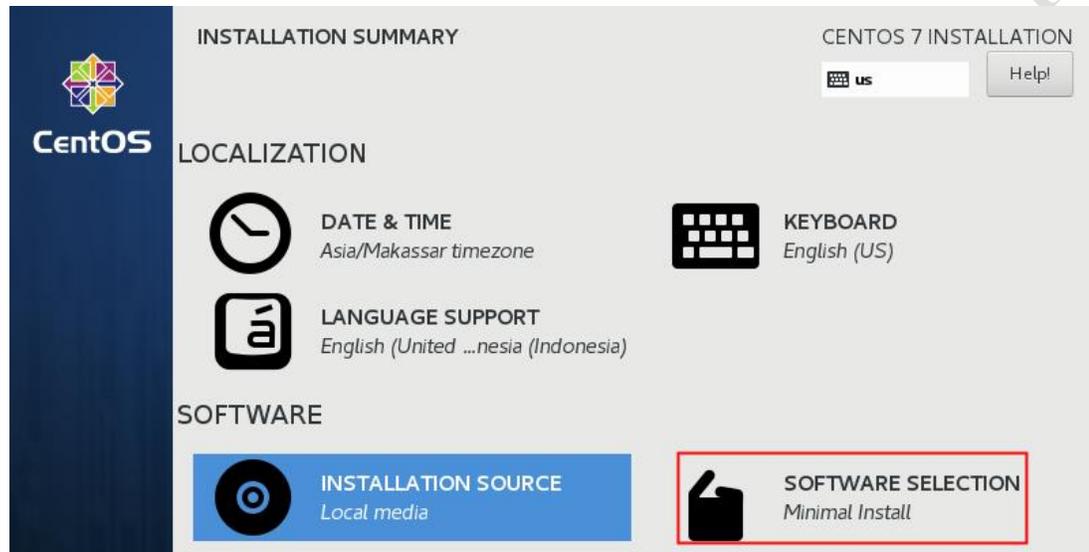
Terlihat saat ini sumber instalasi menggunakan *Local Media*. Apabila ingin dilakukan penyesuaian sumber instalasi yang digunakan maka dapat dilakukan dengan cara klik pada **Installation Source**. Selanjutnya akan tampil pilihan sumber instalasi mana yang ingin digunakan, seperti terlihat pada gambar berikut:



Terlihat terdapat pilihan penggunaan sumber instalasi di jaringan (*network*). Saat ini pilihan tersebut tidak dapat digunakan karena konfigurasi jaringan belum dilakukan.

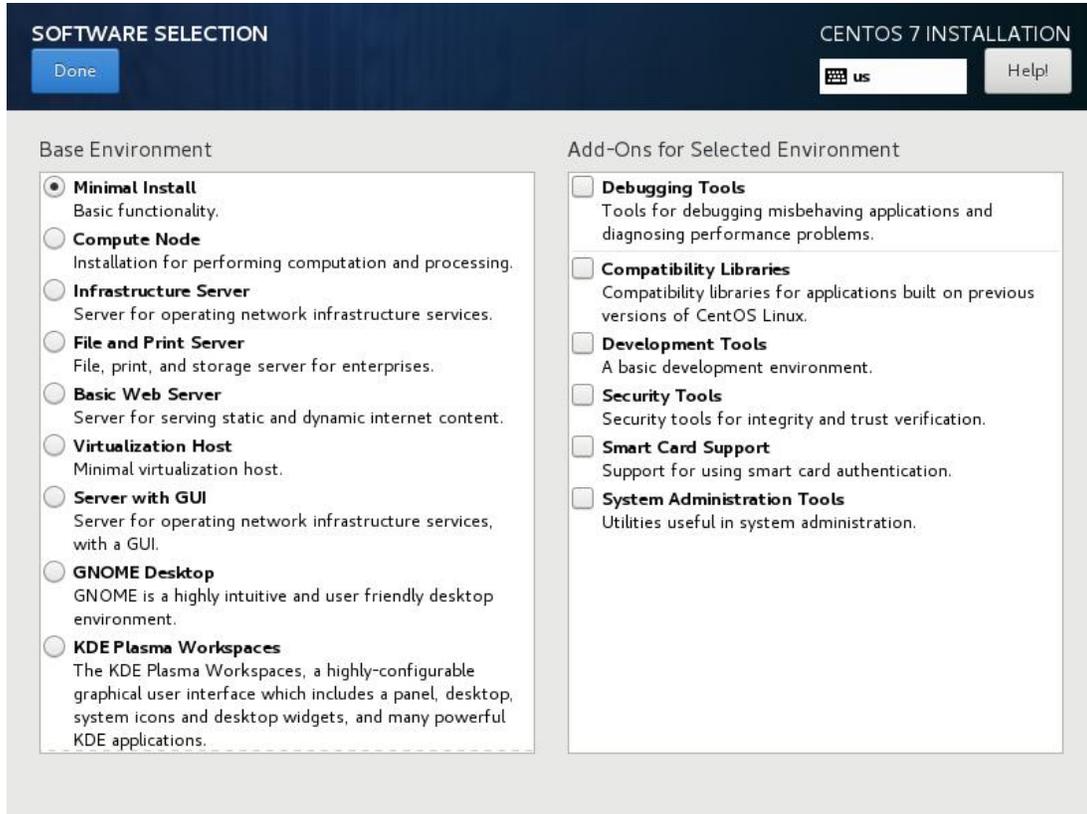
Klik tombol **Done** untuk melanjutkan.

20. Pada layar *Installation Summary* bagian *Software*, terdapat pilihan **Software Selection** untuk mengatur pilihan perangkat lunak yang akan diinstalasi, seperti terlihat pada gambar berikut:



Terlihat saat ini pilihan perangkat lunak yang akan diinstalasi adalah *Minimal Install*.

Penyesuaian dapat dilakukan dengan cara klik pada **Software Selection** maka selanjutnya akan tampil layar dengan pilihan **Base Environment** dan **Add-Ons for Selected Environment**, seperti terlihat pada gambar berikut:



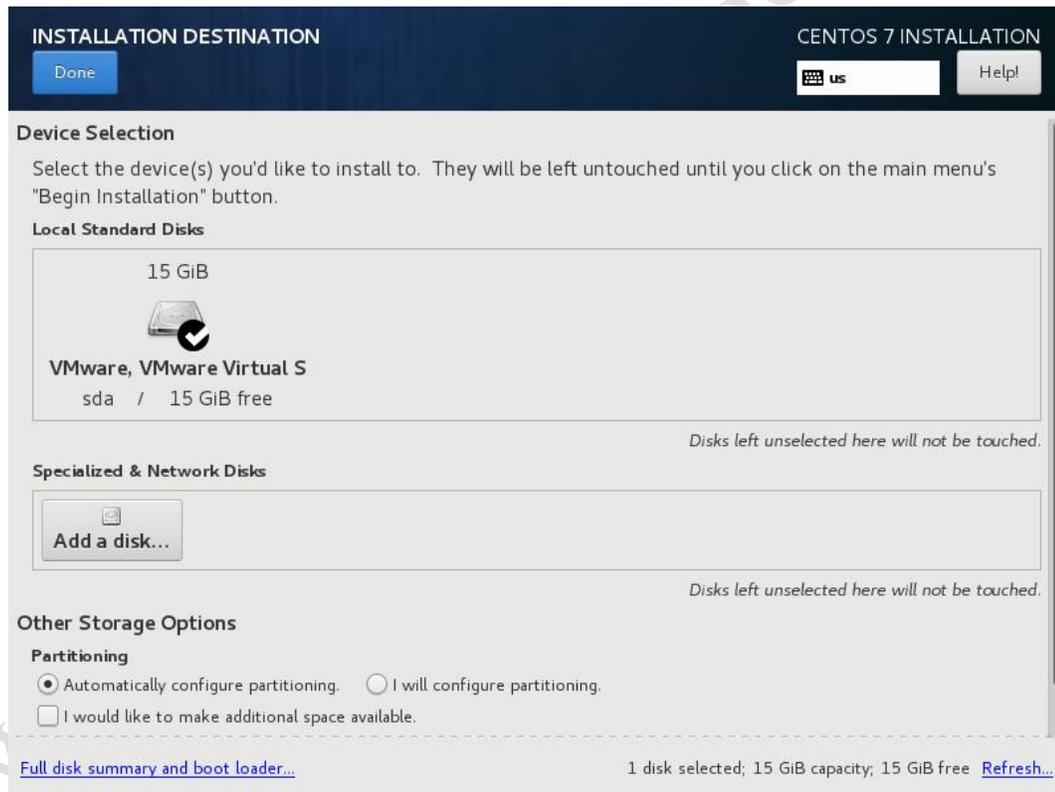
Terdapat 10 (sepuluh) pilihan *Base Environment* atau lingkungan dasar dari perangkat lunak yang dapat digunakan yaitu meliputi *Minimal Install*, *Compute Node*, *Infrastructure Server*, *File and Print Server*, *Basic Web Server*, *Virtualization Host*, *Server with GUI*, *GNOME Desktop*, *KDE Plasma Workspace* dan *Development and Creative Workstation*. Selain itu juga terdapat *Add-Ons* atau tambahan pilihan kelompok perangkat lunak yang dapat diinstalasi bersama dengan *Base Environment* terpilih.

Klik tombol **Done** untuk melanjutkan.

21. Pada layar *Installation Summary* bagian *System*, terdapat pilihan **Instalation Destination** untuk mengatur media atau disk sebagai tujuan instalasi dan pilihan proses partisi, seperti terlihat pada gambar berikut:



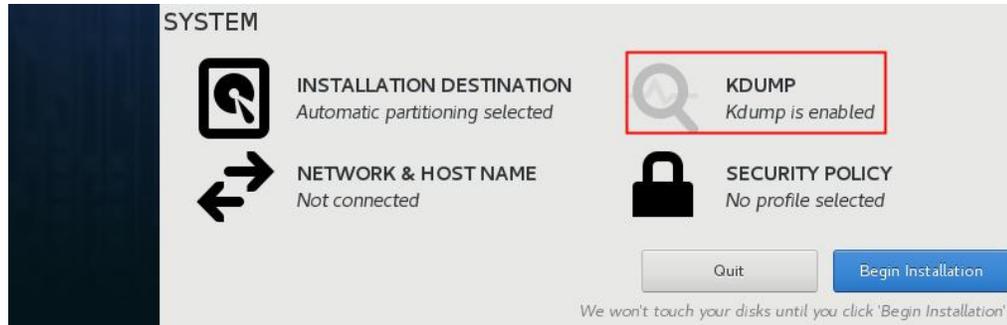
Terlihat saat ini pilihan dari tujuan instalasi adalah *Automatic partitioning selected*. Penyesuaian dapat dilakukan dengan cara klik pada **Installation Destination** maka akan tampil layar, seperti terlihat pada gambar berikut:



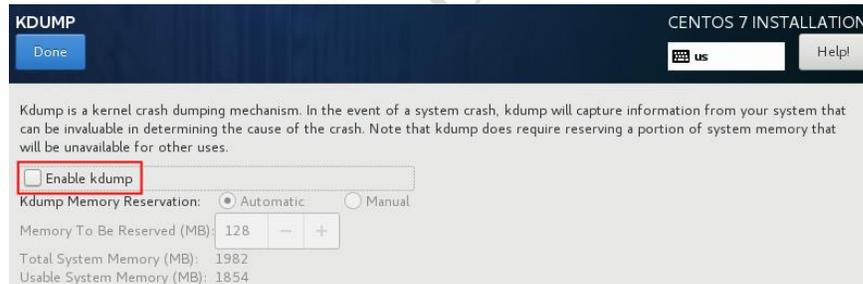
Terlihat pada bagian *Local Standard Disks* telah terseleksi hardisk dengan nama pengenal **sda** dengan kapasitas **15 GiB** sebagai lokasi tujuan instalasi. Selain itu terdapat pula pilihan pengaturan partisi meliputi *Automatic configure partitioning* untuk konfigurasi partisi secara otomatis dan *I will configure partitioning* untuk konfigurasi skema partisi lebih detail. Sebagai contoh skema partisi yang dipilih adalah *Automatic configure partitioning*.

Klik tombol **Done** untuk melanjutkan.

22. Pada layar *Installation Summary* bagian *System*, terdapat pilihan **Kdump** yang merupakan mekanisme *kernel crash dumping*, seperti terlihat pada gambar berikut:

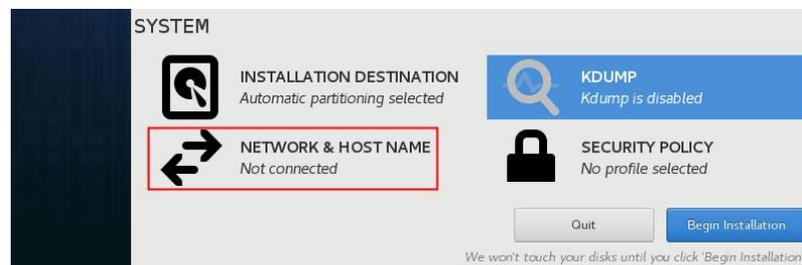


Kdump akan melakukan *capture* informasi dari sistem sehingga dapat menjadi informasi yang berguna untuk menentukan penyebab terjadinya *crash* pada sistem. Terlihat saat ini *Kdump is enabled* yang bermakna fitur ini diaktifkan. Penyesuaian dapat dilakukan dengan cara klik pada **Kdump** maka akan tampil layar untuk menonaktifkan fitur *Kdump*, seperti terlihat pada gambar berikut:

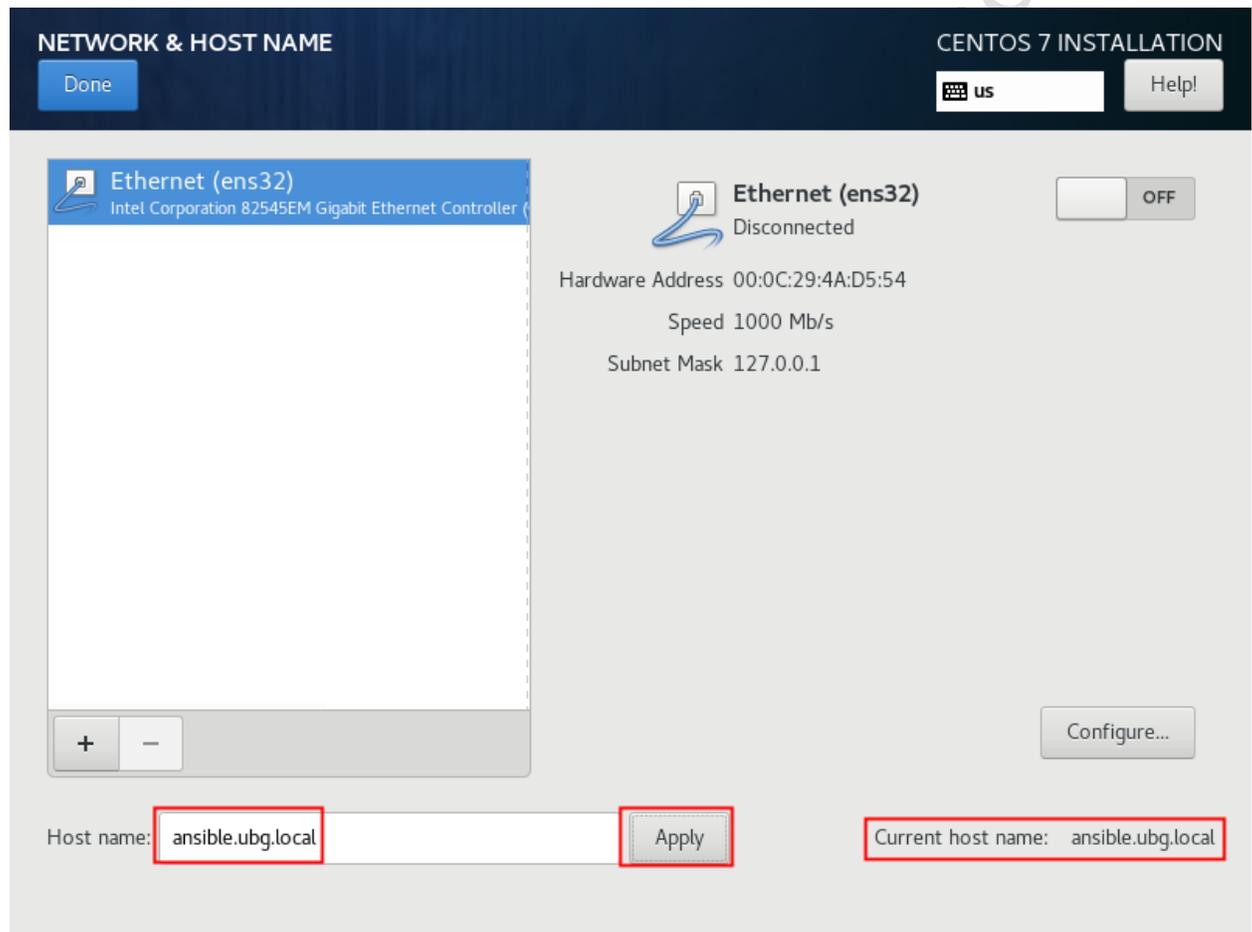


Hilangkan tanda (✓) pada *checkbox* dari parameter *Enable kdump* untuk menonaktifkan fitur tersebut dan klik tombol **Done** untuk melanjutkan.

23. Pada layar *Installation Summary* bagian *System*, terdapat pilihan **Network & Host Name** untuk mengatur konfigurasi *interface* jaringan dan nama pengenal dari sistem, seperti terlihat pada gambar berikut:

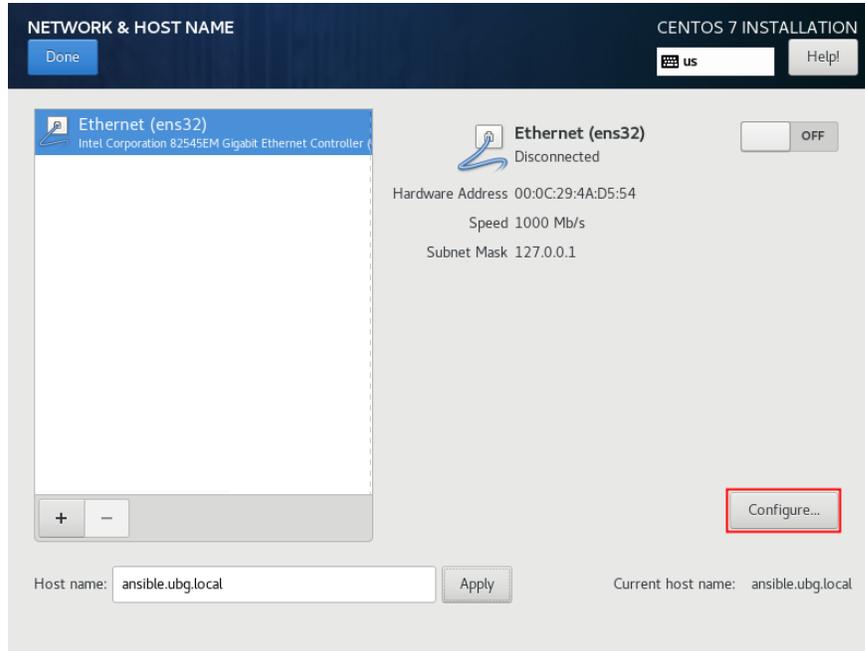


Terlihat saat ini statusnya *Not Connected*. Penyesuaian dapat dilakukan dengan cara klik pada **Network & Host Name** maka akan tampil layar pengaturan **host name:** dan **Ethernet**. Pada inputan parameter **Host name:** masukkan nama pengenal sistem atau komputer. Format nama komputer yang digunakan adalah *Fully Qualified Domain Name (FQDN)* yaitu *namakomputer.namadomain*. Sebagai contoh masukkan nama komputer dengan format FQDN dengan nilai *“ansible.ubg.local”* dan klik tombol **Apply** untuk menyimpan perubahan, seperti terlihat pada gambar berikut:

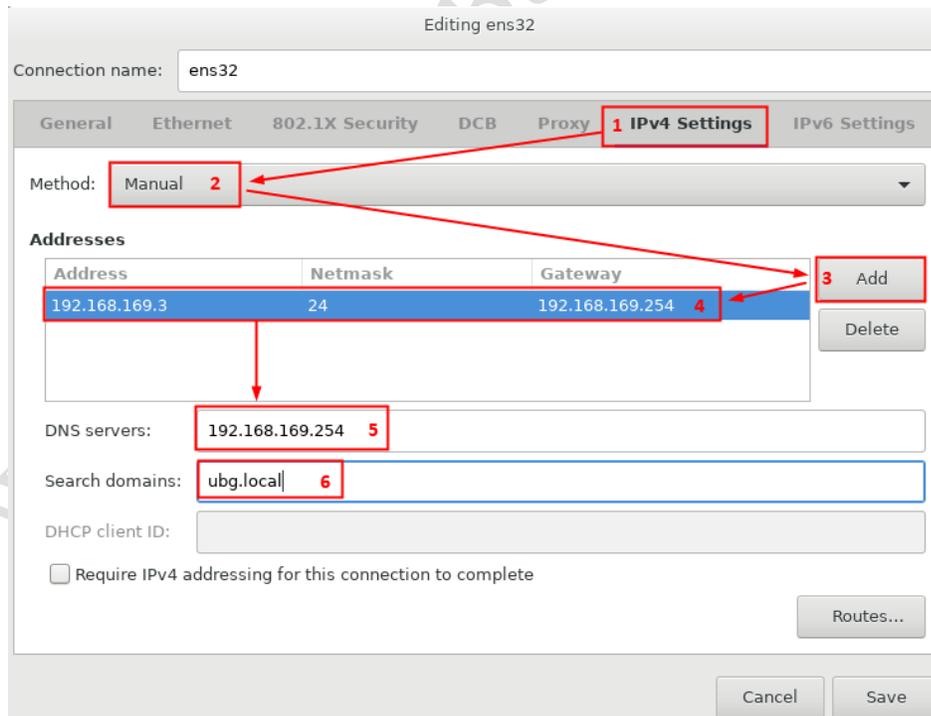


Selain itu terlihat pula keterangan **Current host name:** yang menyatakan nama pengenal sistem yang digunakan saat ini telah berubah dari **localhost** menjadi **ansible.ubg.local**.

Untuk mengatur pengalamatan IP pada interface jaringan **Ethernet (ens32)** dapat dilakukan dengan menekan tombol **Configure**, seperti terlihat pada gambar berikut:



Selanjutnya akan tampil kotak dialog **Editing ens32**. Langkah pertama pilih tab **IPv4 Settings** untuk mengatur pengalamatan IP versi 4, seperti terlihat pada gambar berikut:

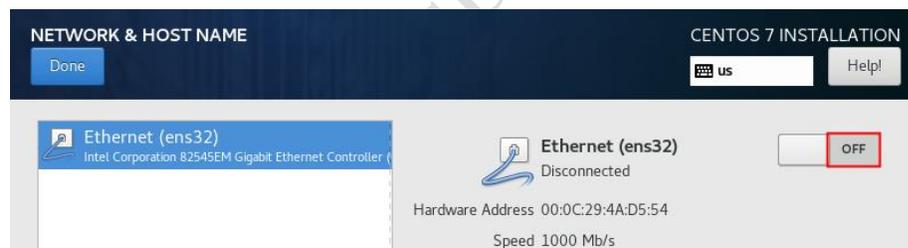


- Langkah kedua, pada *dropdown* parameter **Method**: pilih *Manual* untuk mengatur alokasi pengalamatan IP secara manual.
- Langkah ketiga, klik tombol **Add** untuk menambahkan pengaturan *Addresses*.

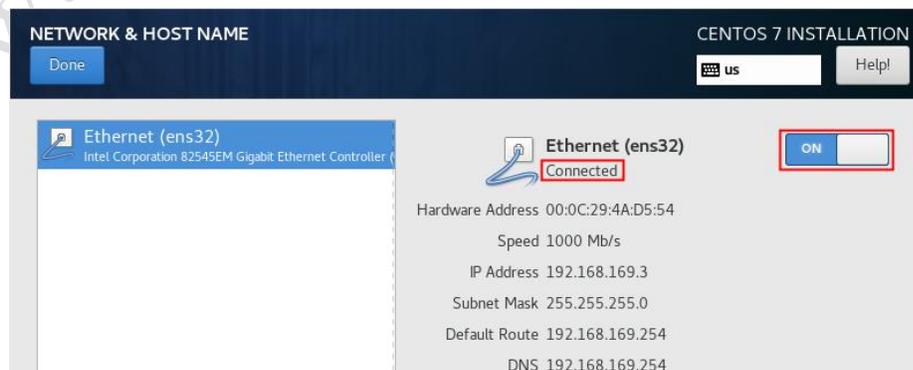
- Langkah ke-empat, lengkapi isian **Addresses** untuk *field Address* yaitu alamat IP yang digunakan, sebagai contoh **192.168.169.3**. *Field Netmask* yaitu alamat subnetmask dari alamat IP yang digunakan dalam format penulisan bit count, sebagai contoh **24**. Sedangkan *field Gateway* yaitu alamat IP dari *router* agar Linux CentOS 7 dapat berkomunikasi ke beda *network* termasuk *Internet*, sebagai contoh **192.168.169.254**.
- Langkah kelima, lengkapi isian **DNS servers:** dengan alamat IP dari server Domain Name System (DNS), sebagai contoh 192.168.169.254.
- Langkah ke-enam, lengkapi isian **Search domains:** dengan nama domain yang digunakan, sebagai contoh *ubg.local*.

Klik tombol **Save** untuk menyimpan perubahan pengaturan interface jaringan.

Selanjutnya mengaktifkan *interface Ethernet (ens32)* dengan cara klik pada **OFF** atau dengan cara menggeser tombol ke sebelah kanan agar menutupi teks **OFF**, seperti terlihat pada gambar berikut:

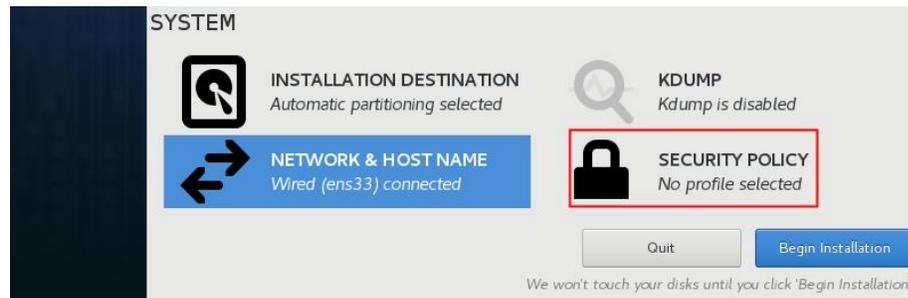


Hasil dari pengaktifan *interface Ethernet (ens32)* menunjukkan status **Ethernet (ens32)** telah berubah dari **Disconnected** menjadi **Connected**, seperti terlihat pada gambar berikut:

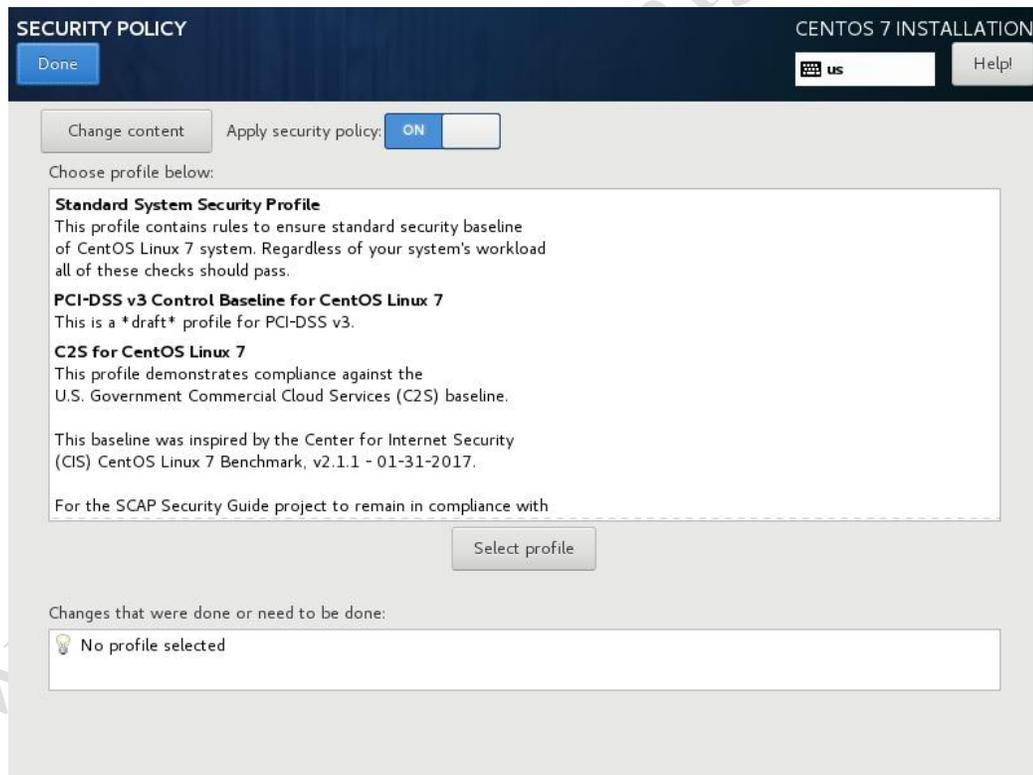


Klik tombol **Done** untuk melanjutkan.

24. Pada layar *Installation Summary* bagian *System*, terdapat pilihan **Security Policy** untuk mengaktifkan *security profile*, seperti terlihat pada gambar berikut:



Terlihat saat ini statusnya *No profile selected* yang menandakan bahwa belum terdapat profil keamanan yang dipilih atau digunakan. Penyesuaian dapat dilakukan dengan cara klik pada **Security Policy** maka akan tampil layar yang memuat pilihan profil, seperti terlihat pada gambar berikut:



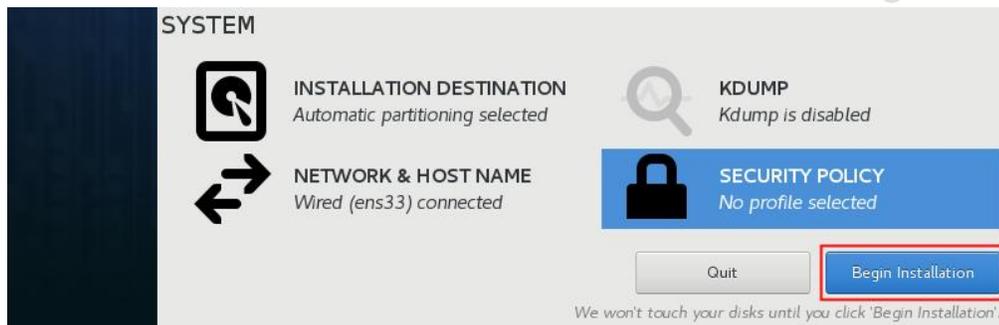
Terdapat beberapa profil yang dapat dipilih meliputi *Standard System Security Profile*, *PCI-DSS v3 Control Baseline for CentOS Linux 7*, *Red Hat Corporate Profile for Certified Cloud Providers*, *Common Profile for General-Purpose Systems*, *Common Profile for General-Purpose Systems*, *United States Government Configuration Baseline (USGCB/STIG)* -

Draft, Criminal Justice Information System (CJIS) Security Policy, Standard Docker Host Profile dan Unclassified Information in Non-federal Information Systems and Organizations (NIST 800-171).

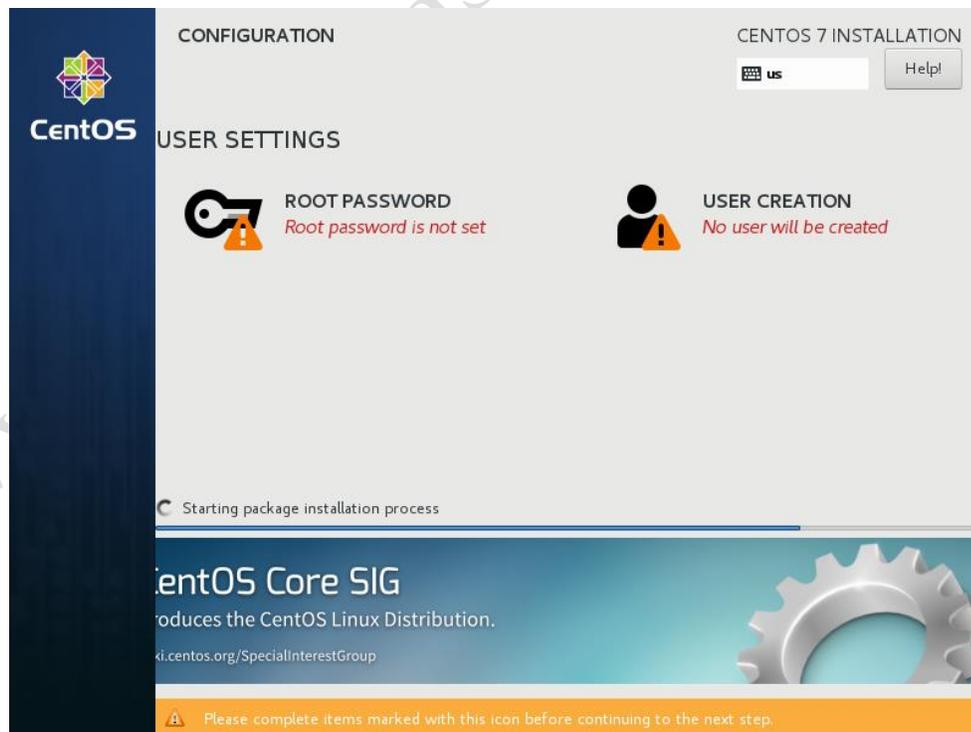
Untuk mengaktifkan *security profile* dapat dilakukan dengan memilih salah satu profil pada bagian *Choose profile below* dan klik tombol **Select profile**.

Klik tombol **Done** untuk melanjutkan.

25. Untuk memulai instalasi maka klik tombol **Begin Installation**, seperti terlihat pada gambar berikut:



Selanjutnya tampil proses instalasi, seperti terlihat pada gambar berikut:

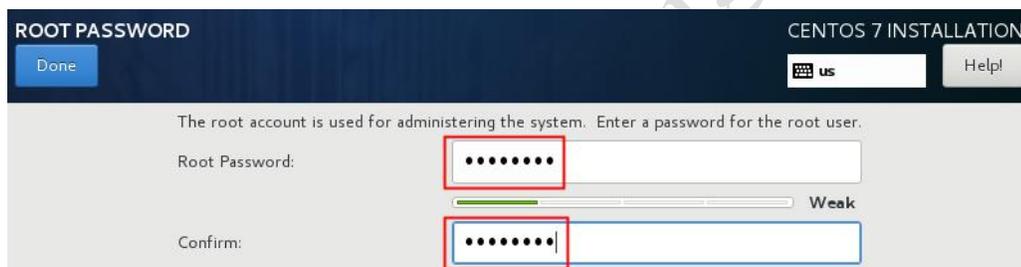


Sambil menunggu proses instalasi berlangsung maka dapat dilakukan pengaturan sandi untuk user root dan pembuatan user baru.

Pengaturan sandi untuk user root dapat dilakukan dengan cara klik pada **Root Password** di bagian *User Settings*, seperti terlihat pada gambar berikut:



Selanjutnya akan tampil layar pengaturan *Root Password*, seperti terlihat pada gambar berikut:



User “**root**” merupakan user tertinggi di Linux, serupa dengan user “**Administrator**” pada sistem operasi Windows. Masukkan sandi yang akan digunakan oleh user “*root*” pada inputan parameter *Root password:* dan ulangi lagi sandi yang dimasukkan pada inputan parameter *Confirm:*, sebagai contoh sandi yang digunakan untuk kebutuhan praktikum adalah “**12345678**”.

Saran untuk keamanan: sebaiknya gunakan sandi yang lebih kompleks seperti minimal memiliki panjang 8 karakter, terdapat minimal 1 huruf kapital, 1 huruf kecil, 1 angka, 1 karakter diluar alfanumerik sehingga sulit untuk ditebak.

Klik tombol **Done** sebanyak dua kali untuk menyimpan perubahan.

Selain itu dapat pula dilakukan pembuatan user baru dengan cara klik pada **User Creation** di bagian *User Settings*, seperti terlihat pada gambar berikut:



Selanjutnya akan tampil layar **Create User**. Terdapat beberapa parameter yang harus dilengkapi, seperti terlihat pada gambar berikut:

 This screenshot shows the 'CREATE USER' screen. It includes a 'Done' button and a 'Help!' button. The form contains the following fields:

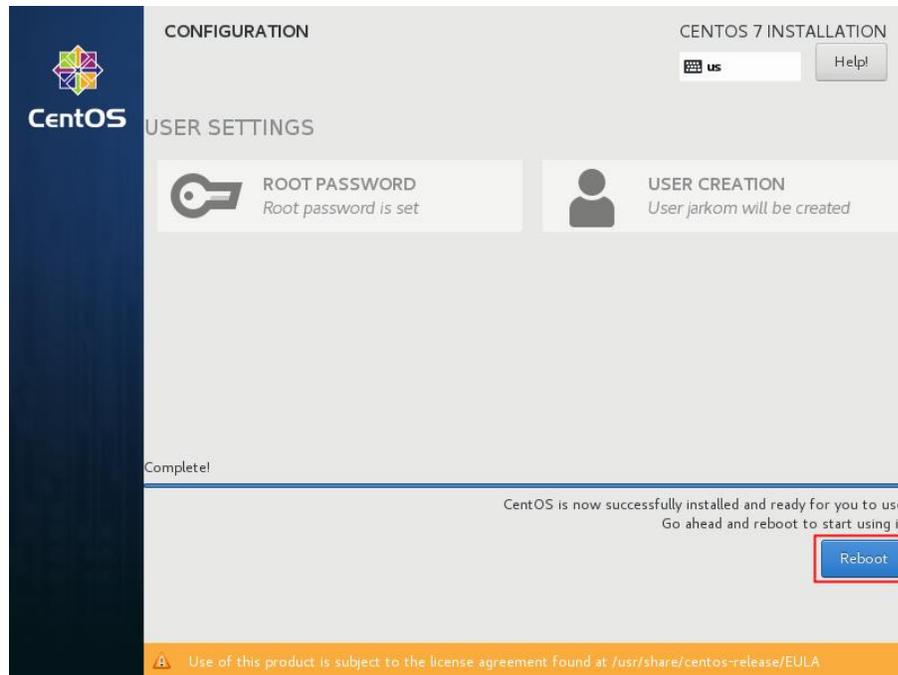
- Full name:** 'Jaringan Komputer' (highlighted with a red box)
- User name:** 'jarkom' (highlighted with a red box)
- Tip:** Keep your user name shorter than 32 characters and do not use spaces.
- Make this user administrator
- Require a password to use this account
- Password:** '.....' (highlighted with a red box)
- Confirm password:** '.....' (highlighted with a red box)
- Strength indicator:** A progress bar showing 'Weak'.
- Advanced...** button

- Full name:*, masukkan nama lengkap pengguna, sebagai contoh **“Jaringan Komputer”**.
- Username:*, masukkan nama login pengguna yang ingin dibuat, sebagai contoh **“jarkom”**.
- Password:*, masukkan sandi login pengguna, sebagai contoh **“12345678”**.
- Confirm Password:*, masukkan kembali sandi login pengguna yaitu **“12345678”**.

Klik tombol **Done** sebanyak dua kali untuk menyimpan.

Tunggu hingga proses instalasi selesai dilakukan.

- Tampil pesan **Complete!** sebagai penanda proses instalasi telah selesai dilakukan, seperti terlihat pada gambar berikut:



Sebelum dapat menggunakan sistem yang telah terinstalasi, Anda diminta untuk melakukan *reboot*. Klik tombol **Reboot**.

27. Setelah proses *booting* selesai maka akan tampil kotak dialog *login* untuk otentikasi awal sebelum dapat menggunakan Linux, seperti terlihat pada gambar berikut:

```
CentOS Linux 7 (Core)
Kernel 3.10.0-957.el7.x86_64 on an x86_64

ansible login:
```

Percobaan login dapat menggunakan user “**root**” atau “**jarkom**”. Sebagai contoh menggunakan user “**root**” maka pada prompt **mataram login:** masukkan “**root**” dan tekan *Enter*. Selanjutnya muncul prompt **Password:** masukkan “**12345678**” dan tekan *Enter*. Apabila proses otentikasi berhasil dilakukan maka akan muncul *prompt terminal Linux*, seperti terlihat pada gambar berikut:

```
CentOS Linux 7 (Core)
Kernel 3.10.0-957.el7.x86_64 on an x86_64

ansible login: root
Password:
Last login: Sat Oct 26 20:01:13 on tty1
[root@ansible ~]# _
```

28. Memverifikasi pengalamatan IP pada *interface* jaringan menggunakan perintah **ip address**, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ip address
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
        valid_lft forever preferred_lft forever
2: ens32: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:4a:d5:54 brd ff:ff:ff:ff:ff:ff
    inet 192.168.169.3/24 brd 192.168.169.255 scope global noprefixroute ens32
        valid_lft forever preferred_lft forever
    inet6 fe80::1597:591f:2c3b:90a4/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Terlihat pengalamatan IP yang digunakan oleh *interface ens32* adalah *192.168.169.3/24*.

29. Memverifikasi *default gateway* dengan menampilkan informasi pada *table routing* menggunakan perintah **ip route**, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ip route
default via 192.168.169.254 dev ens32 proto static metric 100
192.168.169.0/24 dev ens32 proto kernel scope link src 192.168.169.3 metric 100
```

Terlihat pada *table routing* telah terdapat informasi *default gateway* yaitu melalui *192.168.169.254*.

30. Memverifikasi koneksi ke *router Mikrotik CHR* yang berfungsi sebagai *default gateway* menggunakan perintah **ping** ke alamat IP **192.168.169.254**, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ping 192.168.169.254
PING 192.168.169.254 (192.168.169.254) 56(84) bytes of data:
64 bytes from 192.168.169.254: icmp_seq=1 ttl=64 time=0.438 ms
64 bytes from 192.168.169.254: icmp_seq=2 ttl=64 time=0.472 ms
64 bytes from 192.168.169.254: icmp_seq=3 ttl=64 time=0.580 ms
64 bytes from 192.168.169.254: icmp_seq=4 ttl=64 time=1.49 ms
^C
--- 192.168.169.254 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 0.438/0.746/1.494/0.435 ms
```

Terlihat koneksi berhasil dilakukan. Tekan **CTRL+C** untuk menghentikan *ping*.

31. Memverifikasi koneksi *Internet* menggunakan perintah **ping** ke salah satu nama domain di *Internet* sebagai contoh **google.com**, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ping google.com
PING google.com (216.239.38.120) 56(84) bytes of data.
64 bytes from any-in-2678.1e100.net (216.239.38.120): icmp_seq=1 ttl=127 time=35.6 ms
64 bytes from any-in-2678.1e100.net (216.239.38.120): icmp_seq=2 ttl=127 time=35.8 ms
64 bytes from any-in-2678.1e100.net (216.239.38.120): icmp_seq=3 ttl=127 time=47.4 ms
64 bytes from any-in-2678.1e100.net (216.239.38.120): icmp_seq=4 ttl=127 time=52.5 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3013ms
rtt min/avg/max/mdev = 35.681/42.890/52.563/7.338 ms
```

Terlihat koneksi berhasil dilakukan. Tekan **CTRL+C** untuk menghentikan *ping*.

32. Menginstalasi paket aplikasi **nano** sebagai *editor* alternative selain **vi** dengan mengeksekusi perintah **yum -y install nano**, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# yum -y install nano
```

Tunggu hingga proses instalasi selesai dilakukan.

33. Menguji coba penggunaan *editor nano* dengan mengeksekusi perintah:

```
# nano
```

```
GNU nano 2.3.1          New Buffer

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

Terlihat *editor nano* berhasil diakses. Tekan tombol **CTRL+X** untuk keluar dari *editor nano*.

34. Menginstalasi paket *openssh*, *openssh-server*, *openssh-clients* dan *openssl-libs* agar *container* dapat di *remote access* via *SSH Client*.

```
# yum -y install openssh openssh-server openssh-clients
openssl-libs
```

Sehingga terlihat seperti pada gambar berikut:

```
[root@ansible ~]# yum -y install openssh openssh-server openssh-clients openssl-libs
```

Cuplikan hasil instalasi paket, seperti terlihat pada gambar berikut:

```
Updated:
  openssh.x86_64 0:7.4p1-21.e17                openssh-clients.x86_64 0:7.4p1-21.e17
  openssh-server.x86_64 0:7.4p1-21.e17        openssl-libs.x86_64 1:1.0.2k-19.e17

Dependency Updated:
  openssl.x86_64 1:1.0.2k-19.e17

Complete!
[root@ansible ~]# _
```

35. Menjalankan *service sshd* agar layanan *SSH Server* aktif.

```
# systemctl start sshd
```

36. Menampilkan informasi status dari *service sshd* menggunakan perintah:

```
# systemctl status sshd
```

Hasil eksekusi perintah tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# systemctl status sshd
■ sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2019-10-26 21:01:46 WITA; 16s ago
     Docs: man:ssh(8)
           man:ssh_config(5)
   Main PID: 7441 (sshd)
   CGroup: /system.slice/ssh.service
           └─7441 /usr/sbin/sshd -D

Oct 26 21:01:45 ansible.ubg.local systemd[1]: Stopped OpenSSH server daemon.
Oct 26 21:01:45 ansible.ubg.local systemd[1]: Starting OpenSSH server daemon...
Oct 26 21:01:45 ansible.ubg.local sshd[7441]: Server listening on 0.0.0.0 port 22.
Oct 26 21:01:45 ansible.ubg.local sshd[7441]: Server listening on :: port 22.
Oct 26 21:01:46 ansible.ubg.local systemd[1]: Started OpenSSH server daemon.
```

Terlihat *service sshd* telah aktif atau berjalan.

37. Menghentikan *service firewalld* yang merupakan fitur *firewall default* pada *CentOS 7*.

```
# systemctl stop firewalld
```

38. Memverifikasi status *service firewalld*.

```
# systemctl status firewalld
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# systemctl status firewalld
■ firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Sun 2019-10-27 01:07:12 WITA; 7s ago
     Docs: man:firewalld(1)
   Process: 6403 ExecStart=/usr/sbin/firewalld --nofork --nopid $FIREWALLD_ARGS (code=exited, status=0/SUCCESS)
   Main PID: 6403 (code=exited, status=0/SUCCESS)

Oct 27 01:03:42 ansible.ubg.local systemd[1]: Starting firewalld - dynamic firewall daemon...
Oct 27 01:03:44 ansible.ubg.local systemd[1]: Started firewalld - dynamic firewall daemon.
Oct 27 01:07:11 ansible.ubg.local systemd[1]: Stopping firewalld - dynamic firewall daemon...
Oct 27 01:07:12 ansible.ubg.local systemd[1]: Stopped firewalld - dynamic firewall daemon.
```

Terlihat *service firewalld* berhasil dihentikan.

39. Menonaktifkan *service firewalld* secara permanen.

```
# systemctl disable firewalld
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# systemctl disable firewalld
Removed symlink /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
```

40. Memverifikasi penonaktifan *service firewalld*.

```
# systemctl is-enabled firewalld
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# systemctl is-enabled firewalld
disabled
```

Terlihat *service firewalld* telah berhasil dinonaktifkan secara permanen.

41. Menonaktifkan *SELinux* dengan mengubah file `/etc/selinux/config` menggunakan editor *nano*.

```
# nano /etc/selinux/config
```

Lakukan perubahan nilai dari parameter **SELINUX** dengan nilai awal **enforcing**, seperti terlihat pada gambar berikut:

```

GNU nano 2.3.1      File: /etc/selinux/config
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of three values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted

```

Menjadi **disabled**, seperti terlihat pada gambar berikut:

```

GNU nano 2.3.1      File: /etc/selinux/config
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted

```

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan tombol **Enter**.

Keluar dari *editor nano* dengan menekan tombol **CTRL+X**.

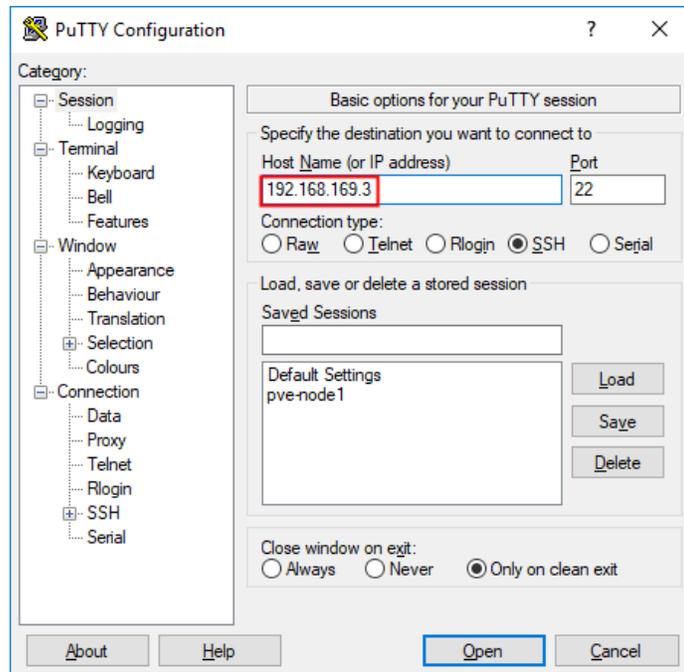
42. Melakukan *reboot* agar penonaktifan *SELinux* berhasil diterapkan.

```
# reboot
```

B. INSTALASI DAN KONFIGURASI ANSIBLE PADA CENTOS 7 SEBAGAI CONTROL MACHINE

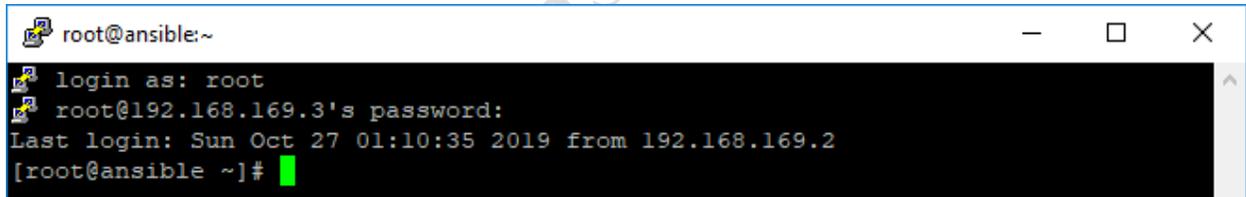
Adapun langkah-langkah instalasi dan konfigurasi *ansible* pada *CentOS 7* sebagai *control machine* adalah sebagai berikut:

1. Jalankan aplikasi *SSH Client*, sebagai contoh menggunakan *Putty*. Tampil kotak dialog *Putty Configuration*. Pada isian **Host Name (or IP Address)**, masukkan alamat IP dari *CentOS 7* yaitu **192.168.169.3**, seperti terlihat pada gambar berikut:



Klik tombol **Open**.

2. Tampil kotak dialog *Putty* yang meminta pengguna untuk melakukan proses otentikasi login ke *CentOS 7*, seperti terlihat pada gambar berikut:

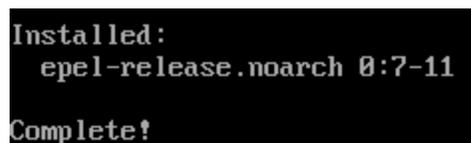


Pada inputan **login as:**, masukkan “**root**” dan tekan tombol **Enter**. Selanjutnya tampil inputan **password:**, masukkan “**12345678**” dan tekan tombol **Enter**. Apabila proses otentikasi berhasil dilakukan maka akan tampil *shell prompt #*.

3. Menginstalasi *repository Extra Packages for Enterprise Linux (EPEL)* karena *ansible* tidak secara *default* terdapat pada **Yum Repositories**.

```
# yum -y install epel-release
```

Cuplikan hasil dari proses instalasi paket EPEL, seperti terlihat pada gambar berikut:



4. Menginstalasi **Ansible**.

```
# yum -y install ansible
```

Cuplikan hasil dari proses instalasi *ansible*, seperti terlihat pada gambar berikut:

```
Installed:
  ansible.noarch 0:2.8.5-1.el7

Dependency Installed:
  PyYAML.x86_64 0:3.10-11.el7
  libyaml.x86_64 0:0.1.4-11.el7_0
  python-babel.noarch 0:0.9.6-8.el7
  python-backports.x86_64 0:1.0-8.el7
  python-backports-ssl_match_hostname.noarch 0:3.5.0.1-1.el7
  python-cffi.x86_64 0:1.6.0-5.el7
  python-enum34.noarch 0:1.0.4-1.el7
  python-httplib2.noarch 0:0.9.2-1.el7
  python-idna.noarch 0:2.4-1.el7
  python-ipaddress.noarch 0:1.0.16-2.el7
  python-jinja2.noarch 0:2.7.2-4.el7
  python-markupsafe.x86_64 0:0.11-10.el7
  python-paramiko.noarch 0:2.1.1-9.el7
  python-ply.noarch 0:3.4-11.el7
  python-pycparser.noarch 0:2.14-1.el7
  python-setuptools.noarch 0:0.9.8-7.el7
  python-six.noarch 0:1.9.0-2.el7
  python2-cryptography.x86_64 0:1.7.2-2.el7
  python2-jmespath.noarch 0:0.9.0-3.el7
  python2-pyasn1.noarch 0:0.1.9-7.el7
  sshpass.x86_64 0:1.06-2.el7

Complete!
[root@ansible ~]#
```

- Menampilkan informasi versi *ansible* yang telah terinstalasi.

```
# ansible --version
```

Hasil dari eksekusi perintah tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible --version
ansible 2.8.5
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/site-packages/ansible
  executable location = /bin/ansible
  python version = 2.7.5 (default, Oct 30 2018, 23:45:53) [GCC 4.8.5 20150623 (Red Hat 4.8.5-36)]
```

Terlihat versi *ansible* yang berhasil diinstalasi adalah **2.8.5**.

- Mengatur *public key authentication* untuk SSH dengan membuat **key pair** menggunakan utilitas **ssh-keygen** pada **CentOS 7 Control Machine** sehingga dapat mengakses **Managed Machine** yaitu **Server PVE** melalui **SSH**.

```
# ssh-keygen
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
```

Tekan **Enter** pada inputan “Enter file in which to save the key (/root/.ssh/id_rsa):” untuk menggunakan lokasi dan nama *file default* sebagai penyimpanan *key* yaitu **/root/.ssh/id_rsa**.

```
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

Tekan **Enter** pada inputan “Enter passphrase (empty for no passphrase):” dan “Enter same passphrase again:” untuk mengosongkan **passphrase**. Selanjutnya akan terlihat proses pembuatan *key*. Tunggu hingga proses selesai dilakukan.

- Menyalinkan *public key* dari **CentOS 7 control machine** ke **PVE managed machine** menggunakan utilitas `ssh-copy-id`.

```
# ssh-copy-id -i root@192.168.169.1
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ssh-copy-id -i root@192.168.169.1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '192.168.169.1 (192.168.169.1)' can't be established.
ECDSA key fingerprint is SHA256:RUyAN/FQjwwOiQQ6TNn3pswBPvaxNxcCSpwFI8Ky6T4.
ECDSA key fingerprint is MD5:4d:fl:eb:de:72:69:9d:c9:b5:5b:ed:65:3a:27:9d:38.
Are you sure you want to continue connecting (yes/no)? yes
/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 prompt
ed now it is to install the new keys
root@192.168.169.1's password:
```

Ketik **yes** dan tekan **Enter** pada pesan konfirmasi “Are you sure you want to continue connection (yes/no)?”.

Selanjutnya tampil inputan **password** untuk *user root* dari **managed machine server PVE** dengan alamat IP **192.168.169.1**. Masukkan “**12345678**” dan tekan **Enter**.

Terlihat *output* hasil akhir proses penyalinan, seperti terlihat pada gambar berikut:

```
Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'root@192.168.169.1'"
and check to make sure that only the key(s) you wanted were added.
```

- Memverifikasi hasil pengaturan *public key authentication* untuk SSH dengan cara mengakses *managed machine* yaitu **server PVE** menggunakan *SSH Client*.

```
# ssh root@192.168.169.1
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ssh root@192.168.169.1
Linux pve 4.15.18-9-pve #1 SMP PVE 4.15.18-30 (Thu, 15 Nov 2018 13:32:46 +0100)
x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Oct 26 10:04:01 2019 from 192.168.169.2
root@pve:~#
```

Server PVE berhasil diakses dari **CentOS7** melalui SSH tanpa **password**. Untuk keluar dari *server PVE*, ketik "**exit**" dan tekan **Enter**, seperti terlihat pada gambar berikut:

```
root@pve:~# exit
logout
Connection to 192.168.169.1 closed.
[root@ansible ~]#
```

- Keluar dari SSH.

```
# exit
```

BAB X

**ANSIBLE INVENTORY, AD-HOC DAN PLAYBOOK SERTA ANSIBLE
PROXMOX_TEMPLATE MODULE**

Bab ini terdiri dari 4 (empat) bagian yaitu *Ansible Inventory*, *Ad-Hoc* dan *Playbook* serta studi kasus penggunaan *Ansible Proxmox_Template Module*.

A. ANSIBLE INVENTORY

Inventory merupakan file inisialisasi yang digunakan oleh *Ansible* untuk mendaftarkan dan mengelompokkan mesin atau *host* yang akan dikelola (*managed machine*). Lokasi dari file *inventory* secara *default* adalah **“/etc/ansible/hosts”** dan memiliki format seperti file INI atau *initialization file format*. Konten di dalam *file inventory* dapat memuat *hostname* atau alamat IP. Contoh sederhana dari konten *file hosts* adalah sebagai berikut:

```
# cat /etc/ansible/hosts
192.168.169.254
pve.ubg.local
```

Terlihat terdapat 2 (dua) *host* yaitu *192.168.169.254* dan *pve.ubg.local*.

Selain itu dapat pula diterapkan pengelompokan (*group*) dimana penulisannya diawali pada bagian atas dengan nama *group* yang diapit dengan tanda kurung siku buka tutup, `[groupname]` dan pada baris berikutnya diikuti oleh *host* yang tergabung pada kelompok tersebut. Menurut dokumentasi [ansible](#), *group* digunakan untuk mengklasifikasi sistem dan menentukan sistem yang dikontrol berdasarkan waktu dan tujuan tertentu. Berikut adalah contoh dari *file hosts* yang telah menerapkan pengelompokan (*group*):

```
# cat /etc/ansible/hosts
[webservers]
www01.universitاسbumigora.ac.id
www02.universitاسbumigora.ac.id
[routers]
```

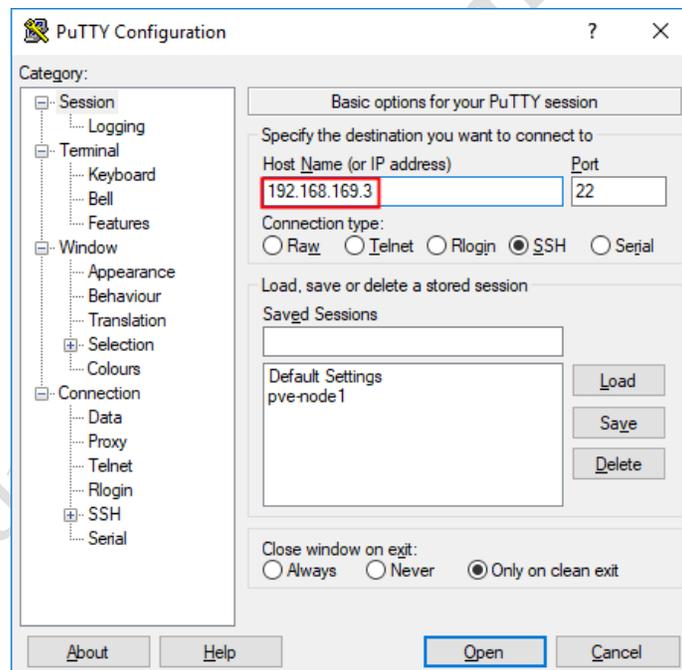
192.168.169.254

192.168.170.254

Terlihat terdapat 2 (dua) *group* yaitu **webserver** dan **routers**. *Group webserver* memiliki 2 (dua) *host* yang ditulis menggunakan format *Fully Qualified Domain Name (FQDN)* yaitu *www01.universitasbumigora.ac.id* dan *www02.universitasbumigora.ac.id*. Begitu pula *group routers* juga memiliki 2 (dua) *host* yang dituliskan menggunakan alamat IP yaitu *192.168.169.254* dan *192.168.170.254*.

Adapun langkah-langkah untuk mengatur *inventory* pada *CentOS 7* yang bertindak sebagai *Ansible Control Machine* adalah sebagai berikut:

1. Jalankan aplikasi *SSH Client*, sebagai contoh menggunakan *Putty*. Tampil kotak dialog *Putty Configuration*. Pada isian **Host Name (or IP Address)**, masukkan alamat IP dari *CentOS 7* yaitu **192.168.169.3**, seperti terlihat pada gambar berikut:



Klik tombol **Open**.

2. Tampil kotak dialog *Putty* yang meminta pengguna untuk melakukan proses otentikasi login ke *CentOS 7*, seperti terlihat pada gambar berikut:

```

root@ansible:~
login as: root
root@192.168.169.3's password:
Last login: Sun Oct 27 01:10:35 2019 from 192.168.169.2
[root@ansible ~]#

```

Pada inputan **login as:**, masukkan “**root**” dan tekan tombol **Enter**. Selanjutnya tampil inputan **password:**, masukkan “**12345678**” dan tekan tombol **Enter**. Apabila proses otentikasi berhasil dilakukan maka akan tampil *shell prompt #*.

- Menambahkan *managed machine* yaitu **server PVE** dengan alamat IP **192.168.169.1** ke baris terbawah pada file **/etc/ansible/hosts**.

```
# nano /etc/ansible/hosts
```

Dengan konten *file hosts*, seperti terlihat pada gambar berikut:

```

GNU nano 2.3.1      File: /etc/ansible/hosts      Modified

# Ex 3: A collection of database servers in the 'dbservers' group

## [dbservers]
##
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57

# Here's another example of host ranges, this time there are no
# leading 0s:

## db-[99:101]-node.example.com

[servers]
pve ansible_host=192.168.169.1

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page  ^U UnCut Text ^T To Spell

```

Penjelasan:

- [servers] merupakan nama pengelompokan atau *group* yaitu **servers**.
- pve merupakan *inventory_hostname* atau nama *alias* bagi *host* dengan alamat IP **192.168.169.1** yang menjadi nilai dari *variable* `ansible_host`. `ansible_host` merupakan *variable* yang dapat digunakan oleh *ansible* untuk melakukan koneksi dengan *remote host* alias `pve`.

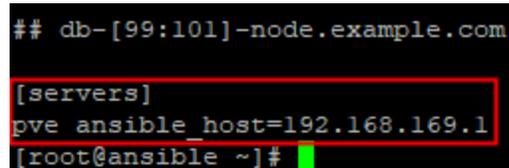
Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan tombol **Enter**.

Keluar dari *editor nano* dengan menekan tombol **CTRL+X**.

4. Memverifikasi hasil penambahan *inventory*.

```
# cat /etc/ansible/hosts
```

Cuplikan hasil dari eksekusi perintah tersebut, seperti terlihat pada gambar berikut:



```
## db-[99:101]-node.example.com
[servers]
pve ansible_host=192.168.169.1
[root@ansible ~]#
```

Terlihat *inventory* terkait *server pve* berhasil ditambahkan.

B. ANSIBLE AD-HOC COMMANDS

Menurut dokumentasi [Ansible](#), **Ad-Hoc command** merupakan salah satu cara dalam menjalankan *ansible* sehingga dapat digunakan untuk melakukan sesuatu dengan cepat dan pengguna tidak berkeinginan untuk menyimpannya ke dalam *playbook*. *Ansible Ad-Hoc command* memiliki sintak penulisan:

```
# ansible <inventory> -m
```

Penjelasan parameter:

- *<inventory>*, menentukan lokasi dari *file inventory host* yang akan digunakan
- *-m*, menentukan nama *module ansible* yang akan digunakan.

Adapun beberapa contoh penggunaan perintah *ansible ad-hoc command* adalah sebagai berikut:

1. Memverifikasi koneksi dan *python* yang digunakan serta terkonfigurasi pada *managed machine* dengan nama *group servers* menggunakan *module ping*.

```
# ansible servers -m ping
```

Hasil eksekusi perintah tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible servers -m ping
[WARNING]: Platform linux on host pve is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.8/referen
ce_appendices/interpreter_discovery.html for more information.

pve | SUCCESS => (
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
)
```

Output menunjukkan bahwa verifikasi koneksi ke *host pve* yang tergabung dalam *group servers* berhasil dilakukan dimana ditandai dengan pesan **pong**. Selain itu terlihat pula informasi *interpreter python* ditemukan pada *host pve* yaitu berada di **/usr/bin/python**.

2. Menampilkan informasi *system uptime* dari *managed machine* dengan nama *group servers* menggunakan *module command*.

```
# ansible servers -m command -a "uptime"
```

Option `-a` digunakan untuk mencantumkan argumen. Hasil eksekusi dari perintah tersebut adalah sebagai berikut:

```
[root@ansible ~]# ansible servers -m command -a "uptime"
[WARNING]: Platform linux on host pve is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.8/referen
ce_appendices/interpreter_discovery.html for more information.

pve | CHANGED | rc=0 >>
11:00:56 up 30 min,  1 user,  load average: 0.00, 0.02, 0.05
```

Output menunjukkan bahwa *host pve* yang tergabung dalam *group servers* telah aktif selama 30 menit dan terdapat satu pengguna yang sedang aktif.

3. Menampilkan informasi ukuran memori baik yang *free* maupun *used* dalam satuan *megabytes* dari *managed machine group "servers"* menggunakan *module command*.

```
# ansible servers -m command -a "free -mt"
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible servers -m command -a "free -mt"
[WARNING]: Platform linux on host pve is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.8/referen
ce_appendices/interpreter_discovery.html for more information.

pve | CHANGED | rc=0 >>
      total      used      free      shared  buff/cache   available
Mem:    1993      690      998         30         304       1123
Swap:   4095         0     4095
Total:  6089      690     5094
```

4. Melakukan **reboot** pada *managed machine* dari group “**servers**” dengan menggunakan *module shell*.

```
# ansible servers -m shell -a "reboot"
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible servers -m shell -a "reboot"
pve | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: Shared connection to 192.168.
169.1 closed.",
  "unreachable": true
}
```

Output menunjukkan bahwa *host pve* yang tergabung dalam *group servers* tidak dapat diakses via SSH karena sedang *reboot*.

C. PLAYBOOK

Merupakan *file* dimana kode *ansible* ditulis dalam format YAML. Setiap *playbook* dapat terdiri dari satu atau lebih *play* didalamnya. *Play* berfungsi untuk memetakan sekumpulan instruksi yang didefinisikan terhadap *host* tertentu. Setiap *play* memuat daftar atau *list* dari *task* yang dieksekusi secara berurutan yaitu satu *task* dalam satu waktu pada semua *host* yang sesuai sebelum menuju ke *task* berikutnya. *Playbook* dieksekusi secara *top-down* sehingga ketika terjadi kegagalan eksekusi *task* pada *host* tertentu maka *playbook* akan berhenti. Untuk itu diperlukan eksekusi ulang *playbook* setelah selesai diperbaiki.

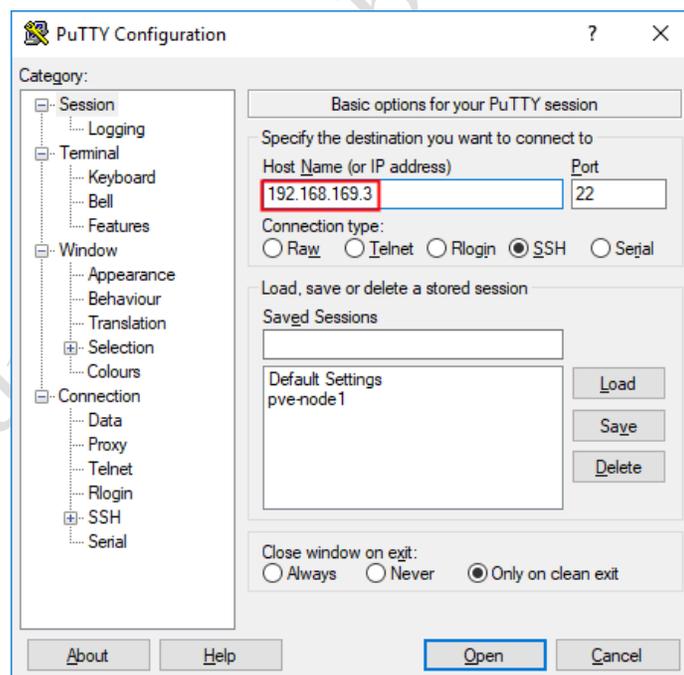
Menurut situs [Tutorialspoint](https://www.tutorialspoint.com/ansible/ansible-playbooks.htm), YAML memiliki beberapa *tags* yang umum digunakan pada *playbook* yaitu antara lain:

- a) *name* digunakan untuk menentukan nama dari *ansible playbook*.

- b) `hosts` digunakan untuk menentukan daftar *host* atau *group host* sebagai lokasi eksekusi *task*.
- c) `vars` digunakan untuk mendefinisikan *variable* playbook dimana penggunaannya serupa dengan yang terdapat di bahasa pemrograman.
- d) `tasks` merupakan daftar tindakan atau aksi yang perlu dieksekusi. Didalam *tasks* memuat nama dari *task* dan kode yang berhubungan dengan modul yang harus dieksekusi.

Adapun contoh sederhana dari pembuatan dan verifikasi serta eksekusi *ansible playbook* adalah sebagai berikut:

1. Jalankan aplikasi *SSH Client*, sebagai contoh menggunakan *PuTTY*. Tampil kotak dialog *PuTTY Configuration*. Pada isian **Host Name (or IP Address)**, masukkan alamat IP dari *CentOS 7 Ansible Control Machine* yaitu **192.168.169.3**, seperti terlihat pada gambar berikut:



Klik tombol **Open**.

2. Tampil kotak dialog *PuTTY* yang meminta pengguna untuk melakukan proses otentikasi login ke *CentOS 7*, seperti terlihat pada gambar berikut:

```

root@ansible:~
login as: root
root@192.168.169.3's password:
Last login: Sun Oct 27 01:10:35 2019 from 192.168.169.2
[root@ansible ~]#

```

Pada inputan **login as:**, masukkan “**root**” dan tekan tombol **Enter**. Selanjutnya tampil inputan **password:**, masukkan “**12345678**” dan tekan tombol **Enter**. Apabila proses otentikasi berhasil dilakukan maka akan tampil *shell prompt #*.

3. Membuat file *playbook* dengan nama “*basic.yml*” menggunakan *editor nano*. *Playbook* dari contoh ini dapat diunduh pada situs **Github** dengan alamat <https://github.com/iputuhariyadi/ansible-proxmox-playbook>.

```
# nano basic.yml
```

Dengan konten seperti berikut:

```

1---
2- name: Dasar YAML
3  hosts: servers
4  vars:
5    pesan: "Hello World"
6  tasks:
7    - name: Menampilkan Variable
8      debug: var=pesan

```

Penjelasan:

- Baris 1: `---` (3 *hyphen*) merupakan awal dari dokumen **YAML**.
- Baris 2: `name` digunakan untuk menentukan nama dari *ansible playbook* yaitu “**Dasar YAML**”.
- Baris 3: `hosts` digunakan untuk menentukan daftar *host* atau *group host* sebagai lokasi eksekusi *task* yaitu *group servers*.

- Baris 4: *vars* digunakan untuk mendefinisikan *variable* pada *playbook* dimana penggunaannya serupa dengan yang terdapat di bahasa pemrograman.
- Baris 5: Deklarasi *variable* dengan nama “**pesan**” yang memiliki nilai “**Hello World**”.
- Baris 6: *tasks* merupakan daftar tindakan atau aksi yang perlu dieksekusi.
- Baris 7: *name* digunakan untuk menentukan nama dari *task* yaitu “**Menampilkan Variable**”. Di dalam setiap *task* memuat kode yang berhubungan dengan *modul* yang harus dieksekusi.
- Baris 8: nama modul yang dieksekusi yaitu **debug** yang berguna untuk menampilkan pesan ketika eksekusi. Modul ini memiliki parameter **var** yang digunakan untuk menentukan nama *variable* yang di *debug* yaitu **pesan**.

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan tombol **Enter**.

Keluar dari *editor nano* dengan menekan tombol **CTRL+X**.

4. Mengecek sintak dari *playbook* sebelum dieksekusi sehingga memastikan tidak terdapat permasalahan terkait sintak dengan mengeksekusi perintah `ansible-playbook` dengan flag `--syntax-check`.

```
# ansible-playbook basic.yml --syntax-check
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible-playbook basic.yml --syntax-check
playbook: basic.yml
```

5. Menampilkan informasi host yang terdampak oleh *playbook* sebelum dieksekusi dapat menggunakan perintah:

```
# ansible-playbook basic.yml --list-hosts
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible-playbook basic.yml --list-host
playbook: basic.yml

play #1 (servers): Dasar YAML TAGS: []
  pattern: [u'servers']
  hosts (1):
    pve
```

Terlihat bahwa *host* **pve** pada *group* **servers** yang terdampak oleh eksekusi *playbook*.

6. Mengeksekusi *playbook*.

```
# ansible-playbook basic.yml
```

Hasil eksekusi dari *playbook* tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible-playbook basic.yml

PLAY [Dasar YAML] **1**
TASK [Gathering Facts] **2**
ok: [pve]

TASK [Menampilkan Variable] **3**
ok: [pve] => {
  "pesan": "Hello World" **4**
}

PLAY RECAP *****
pve : ok=2 **5** changed=0 unreachable=0 failed=0
kipped=0 rescued=0 ignored=0
```

Penjelasan: (1) nama *play* yang dieksekusi yaitu “Dasar YAML”, (2) eksekusi *task gathering facts* yang memuat informasi terkait *remote host* dan terlihat pesan status **ok** yang menyatakan *task* dapat dieksekusi dengan baik, (3) eksekusi *task “Menampilkan Variable”* dan terlihat pesan status **ok** yang menyatakan *task* dapat dieksekusi dengan baik serta menampilkan hasil *debug variable pesan*, (4) metric **ok** menginformasikan rangkuman jumlah *task* yang dapat dieksekusi dengan baik atau tidak mengubah apapun pada **servers**.

D. STUDI KASUS PENGGUNAAN ANSIBLE PROXMOX_TEMPLATE

Playbook dari studi kasus yang dibahas pada sub bab ini dapat diunduh pada situs **Github** dengan alamat <https://github.com/iputuhariyadi/ansible-proxmox-playbook>. Adapun contoh

penggunaan *ansible module proxmox_template* pada *playbook* untuk manajemen *container template* di server PVE adalah sebagai berikut:

1. Mengunggah *container template CentOS 8* dengan nama file “**centos-8-default_20191016_amd64.tar.xz**” ke server PVE.

a) Membuat *playbook* dengan nama “unggah-container-template.yml”.

```
# nano unggah-container-template.yml
```

Dengan konten sebagai berikut:

```
1 ---
2 - name: Manajemen Container Template di PVE
3   hosts: servers
4   tasks:
5     - name: Unggah Container Template
6       proxmox_template:
7         api_host: pve
8         api_user: root@pam
9         api_password: '12345678'
10        node: pve
11        src: ~/centos-8-default_20191016_amd64.tar.xz
```

Penjelasan:

- Baris 1: --- (3 hyphen) merupakan awal dari dokumen YAML.
- Baris 2: *name* digunakan untuk menentukan nama dari *ansible playbook* yaitu “**Manajemen Container Template di PVE**”.
- Baris 3: *hosts* digunakan untuk menentukan daftar *host* atau *group host* sebagai lokasi eksekusi *task* yaitu *group servers*.
- Baris 4: *tasks* merupakan daftar tindakan atau aksi yang perlu dieksekusi.
- Baris 5: *name* digunakan untuk menentukan nama dari *task* yaitu “**Unggah Container Template**”. Di dalam setiap *task* memuat kode yang berhubungan dengan *modul* yang harus dieksekusi.
- Baris 6: nama modul yang dieksekusi yaitu **proxmox_template**.
- Baris 7: merupakan parameter dari *ansible proxmox module* untuk menentukan *host* dari *Proxmox VE Cluster* yaitu “**pve**”.
- Baris 8: merupakan parameter dari *ansible proxmox module* untuk menentukan *user* yang digunakan *login* ke *Proxmox VE* yaitu “**root@pam**”.

- Baris 9: merupakan parameter dari *ansible proxmox module* untuk menentukan *password* otentikasi ke *Proxmox VE* yaitu “12345678”.
- Baris 10: merupakan parameter dari *ansible proxmox module* untuk menentukan *node* dari *Proxmox VE* sebagai lokasi *container template* yaitu **pve**.
- Baris 11: merupakan parameter dari *ansible proxmox module* untuk menentukan *path* dari file yang akan diunggah ke *Proxmox* yaitu di home direktori dari user yang digunakan *login (root)* dengan nama *file* “centos-8-default_20191016_amd64.tar.xz”.

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan tombol **Enter**.

Keluar dari *editor nano* dengan menekan tombol **CTRL+X**.

b) Mengeksekusi playbook.

```
# ansible-playbook unggah-container-template.yml
```

Hasil eksekusi dari playbook, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible-playbook unggah-container-template.yml
PLAY [Manajemen Container Tempate di PVE] *****
TASK [Gathering Facts] *****
ok: [pve]
TASK [Unggah Container Template] *****
changed: [pve]
PLAY RECAP *****
pve : ok=2  changed=1  unreachable=0  failed=0  s
kipped=0  rescued=0  ignored=0
```

c) Memverifikasi *container template CentOS 8* yang telah diunggah menggunakan *ansible ad-hoc command*.

```
# ansible servers -m command -a "pvesh get /nodes/pve/storage/local/content"
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible servers -m command -a "pvesh get /nodes/pve/storage/local/content"
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at
/usr/bin/python, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.8/reference_appendices/interpreter_discovery.html for more
information.

pve | CHANGED | rc=0 >>
```

format	size	valid	parent	used	vmid
txz	65.96 MiB	local:vztmpl/centos-7-default_20171212_amd64.tar.xz			
txz	101.32 MiB	local:vztmpl/centos-8-default_20191016_amd64.tar.xz			

Terlihat *container template CentOS 8* berhasil diunggah ke *local storage* di server PVE.

2. Menghapus *container template CentOS 8* di server PVE.

- a) Membuat *playbook* dengan nama “hapus-container-template.yml” dengan menerapkan *variable* untuk menampung akun otentikasi untuk koneksi ke server PVE.

```
# nano hapus-container-template.yml
```

Dengan konten sebagai berikut:

```
1 ---
2 - name: Manajemen Container Template di PVE
3   hosts: servers
4   vars:
5     api_host: pve
6     api_user: root@pam
7     api_password: '12345678'
8     node: pve
9   tasks:
10  - name: Hapus Container Template
11    proxmox_template:
12      api_host: "{{ api_host }}"
13      api_user: "{{ api_user }}"
14      api_password: "{{ api_password }}"
15      node: "{{ node }}"
16      template: centos-8-default_20191016_amd64.tar.xz
17      state: absent
```

Penjelasan:

- Baris 1: --- (3 hyphen) merupakan awal dari dokumen YAML.
- Baris 2: *name* digunakan untuk menentukan nama dari *ansible playbook* yaitu “Manajemen Container Template di PVE”.
- Baris 3: *hosts* digunakan untuk menentukan daftar *host* atau *group host* sebagai lokasi eksekusi *task* yaitu *group servers*.

- Baris 4: *vars* digunakan untuk mendefinisikan *variable* pada *playbook* dimana penggunaannya serupa dengan yang terdapat di bahasa pemrograman.
- Baris 5-8: Deklarasi 4 (empat) *variable* yaitu “**api_host**” dengan nilai “**pve**”, “**api_user**” dengan nilai “**root@pam**”, “**api_password**” dengan nilai “**12345678**” dan “**node**” dengan nilai “**pve**”.
- Baris 9: *tasks* merupakan daftar tindakan atau aksi yang perlu dieksekusi.
- Baris 10: *name* digunakan untuk menentukan nama dari *task* yaitu “**Hapus Container Template**”. Di dalam setiap *task* memuat kode yang berhubungan dengan *modul* yang harus dieksekusi.
- Baris 11: nama modul yang dieksekusi yaitu **proxmox_template**.
- Baris 12: merupakan parameter dari *ansible proxmox module* untuk menentukan *host* dari *Proxmox VE Cluster* yang nilainya diambil dari *variable* “`{{ api_host }}`”.
- Baris 13: merupakan parameter dari *ansible proxmox module* untuk menentukan *user* yang digunakan *login* ke *Proxmox VE* yang nilainya diambil dari *variable* “`{{ api_user }}`”.
- Baris 14: merupakan parameter dari *ansible proxmox module* untuk menentukan *password* otentikasi ke *Proxmox VE* yang nilainya diambil dari *variable* “`{{ api_password }}`”.
- Baris 15: merupakan parameter dari *ansible proxmox module* untuk menentukan *node* dari *Proxmox VE* sebagai lokasi *container template* yang nilainya diambil dari *variable* “`{{ node }}`”.
- Baris 16: merupakan parameter dari *ansible proxmox module* untuk menentukan nama dari file *template* yang akan dihapus pada *Proxmox* yaitu “`centos-8-default_20191016_amd64.tar.xz`”.
- Baris 17: merupakan parameter dari *ansible proxmox module* untuk menghapus *template*.

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan tombol **Enter**.

Keluar dari *editor nano* dengan menekan tombol **CTRL+X**.

b) Mengeksekusi playbook.

```
# ansible-playbook hapus-container-template.yml
```

Hasil eksekusi dari playbook, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible-playbook hapus-container-template.yml

PLAY [Manajemen Container Template di PVE] *****

TASK [Gathering Facts] *****
ok: [pve]

TASK [Hapus Container Template] *****
changed: [pve]

PLAY RECAP *****
pve                : ok=2    changed=1    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
```

c) Memverifikasi *container template CentOS 8* yang telah dihapus menggunakan *ansible ad-hoc command*.

```
# ansible servers -m command -a "pvesh get /nodes/pve/storage/local/content"
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible servers -m command -a "pvesh get /nodes/pve/storage/local/content"
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at
/usr/bin/python, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.8/reference_appendices/interpreter_discovery.html for more
information.

pve | CHANGED | rc=0 >>
+-----+-----+-----+-----+-----+-----+
| format | size  | valid | parent | used | vmid |
+-----+-----+-----+-----+-----+-----+
| txz    | 65.96 MiB | local:vztmpl/centos-7-default_20171212_amd64.tar.xz |  |  |  |
+-----+-----+-----+-----+-----+-----+
```

Terlihat *container template CentOS 8* telah berhasil dihapus dari *local storage* di server *PVE*.

3. Memanajemen *container template* baik mengunggah maupun menghapus dalam satu *file playbook* dan menggunakan *file* terpisah untuk menampung *variable* terkait akun otentikasi ke server *PVE*.

a) Membuat *file* *YAML* dengan nama "pve_auth_vars.yml" yang memuat *variable* terkait akun otentikasi ke server *PVE*.

```
# nano pve_auth_vars.yml
```

Dengan konten sebagai berikut:

```
1 api_host: pve
2 api_user: root@pam
3 api_password: '12345678'
4 node: pve
```

Penjelasan:

- Baris 1: Deklarasi *variable* “**api_host**” dengan nilai “**pve**”.
- Baris 2: Deklarasi *variable* “**api_user**” dengan nilai “**root@pam**”.
- Baris 3: Deklarasi *variable* “**api_password**” dengan nilai “**12345678**”.
- Baris 4: Deklarasi *variable* “**node**” dengan nilai “**pve**”.

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan tombol **Enter**.

Keluar dari *editor nano* dengan menekan tombol **CTRL+X**.

- b) Membuat *playbook* dengan nama “manajemen-container-template.yml” dan menyisipkan *file variable* “pve_auth_vars.yml” pada bagian *task*. Selain itu juga diterapkan atribut **tags** sehingga ketika *playbook* dieksekusi dapat mem-*filter task* tertentu yang dijalankan yaitu dalam hal *task unggah* atau *hapus container template*.

```
# nano manajemen-container-template.yml
```

Dengan konten sebagai berikut:

```
1 ---
2 - name: Manajemen Container Template di PVE
3   hosts: servers
4   tasks:
5     - name: include vars
6       include_vars: pve_auth_vars.yml
7
8     - name: Unggah Container Template
9       proxmox_template:
10         api_host: "{{ api_host }}"
11         api_user: "{{ api_user }}"
```

```

12     api_password: "{{ api_password }}"
13     node: "{{ node }}"
14     src: ~/centos-8-default_20191016_amd64.tar.xz
15     tags: unggah
16
17 - name: Hapus Container Template
18   proxmox_template:
19     api_host: "{{ api_host }}"
20     api_user: "{{ api_user }}"
21     api_password: "{{ api_password }}"
22     node: "{{ node }}"
23     template: centos-8-default_20191016_amd64.tar.xz
24     state: absent
25     tags: hapus

```

Penjelasan:

- Baris 1: `---` (3 hyphen) merupakan awal dari dokumen YAML.
- Baris 2: `name` digunakan untuk menentukan nama dari *ansible playbook* yaitu **“Manajemen Container Template di PVE”**.
- Baris 3: `hosts` digunakan untuk menentukan daftar *host* atau *group host* sebagai lokasi eksekusi *task* yaitu *group servers*.
- Baris 4: `tasks` merupakan daftar tindakan atau aksi yang perlu dieksekusi.
- Baris 5: `name` digunakan untuk menentukan nama dari *task* yaitu **“include vars”**. Di dalam setiap *task* memuat kode yang berhubungan dengan *modul* yang harus dieksekusi.
- Baris 6: `include_vars` digunakan untuk memuat *variable* dari *file* yaitu `“pve_auth_vars.yml”`.
- Baris 8: `name` digunakan untuk menentukan nama dari *task* yaitu **“Unggah Container Template”**. Di dalam setiap *task* memuat kode yang berhubungan dengan *modul* yang harus dieksekusi.

- Baris 9: nama modul yang dieksekusi yaitu **proxmox_template**.
- Baris 10: merupakan parameter dari *ansible proxmox module* untuk menentukan *host* dari *Proxmox VE Cluster* yang nilainya diambil dari *variable* “`{{ api_host }}`”.
- Baris 11: merupakan parameter dari *ansible proxmox module* untuk menentukan *user* yang digunakan *login* ke *Proxmox VE* yang nilainya diambil dari *variable* “`{{ api_user }}`”.
- Baris 12: merupakan parameter dari *ansible proxmox module* untuk menentukan *password* otentikasi ke *Proxmox VE* yang nilainya diambil dari *variable* “`{{ api_password }}`”.
- Baris 13: merupakan parameter dari *ansible proxmox module* untuk menentukan *node* dari *Proxmox VE* sebagai lokasi *container template* yang nilainya diambil dari *variable* “`{{ node }}`”.
- Baris 14: merupakan parameter dari *ansible proxmox module* untuk menentukan *path* dari file yang akan diunggah ke *Proxmox* yaitu di home direktori dari user yang digunakan *login* (*root*) dengan nama *file* “centos-8-default_20191016_amd64.tar.xz”.
- Baris 15: *tags* untuk menandai task “**Unggah Container Template**” yaitu “**unggah**” sehingga ketika eksekusi *playbook* dapat memfilter *task* tertentu yang dijalankan.
- Baris 17: *name* digunakan untuk menentukan nama dari *task* yaitu “**Hapus Container Template**”. Di dalam setiap *task* memuat kode yang berhubungan dengan *modul* yang harus dieksekusi.
- Baris 18: nama modul yang dieksekusi yaitu **proxmox_template**.
- Baris 19: merupakan parameter dari *ansible proxmox module* untuk menentukan *host* dari *Proxmox VE Cluster* yang nilainya diambil dari *variable* “`{{ api_host }}`”.

- Baris 20: merupakan parameter dari *ansible proxmox module* untuk menentukan *user* yang digunakan *login* ke *Proxmox VE* yang nilainya diambil dari *variable* “`{{ api_user }}`”.
- Baris 21: merupakan parameter dari *ansible proxmox module* untuk menentukan *password* otentikasi ke *Proxmox VE* yang nilainya diambil dari *variable* “`{{ api_password }}`”.
- Baris 22: merupakan parameter dari *ansible proxmox module* untuk menentukan *node* dari *Proxmox VE* sebagai lokasi *container template* yang nilainya diambil dari *variable* “`{{ node }}`”.
- Baris 23: merupakan parameter dari *ansible proxmox module* untuk menentukan nama dari file *template* yang akan dihapus pada *Proxmox* yaitu “`centos-8-default_20191016_amd64.tar.xz`”.
- Baris 24: merupakan parameter dari *ansible proxmox module* untuk menghapus *template*.
- Baris 25: *tags* untuk menandai task “**Hapus Container Template**” yaitu “**hapus**” sehingga ketika eksekusi *playbook* dapat memfilter *task* tertentu yang dijalankan.

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan tombol **Enter**.

Keluar dari *editor nano* dengan menekan tombol **CTRL+X**.

- c) Mengujicoba mengunggah *container template CentOS 8* dengan mengeksekusi *playbook* “`manajemen-container-template.yml`” dan keseluruhan *task* didalamnya kecuali *task* dengan *tags hapus*.

```
# ansible-playbook manajemen-container-template.yml --skip-tags "hapus"
```

Hasil eksekusi dari *playbook* tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible-playbook manajemen-container-template.yml --skip-tags "hapus"
PLAY [Manajemen Container Template di PVE] *****
TASK [Gathering Facts] *****
ok: [pve]
TASK [include vars] *****
ok: [pve]
TASK [Unggah Container Template] *****
changed: [pve]
PLAY RECAP *****
pve                : ok=3   changed=1   unreachable=0   failed=0   skipped=0   rescued=
0   ignored=0
[root@ansible ~]#
```

d) Memverifikasi *container template CentOS 8* yang telah diunggah menggunakan *ansible ad-hoc command*.

```
# ansible servers -m command -a "pvesh get /nodes/pve/storage/local/content"
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible servers -m command -a "pvesh get /nodes/pve/storage/local/content"
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at
/usr/bin/python, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.8/reference_appendices/interpreter_discovery.html for more
information.
pve | CHANGED | rc=0 >>
```

format	size	valid	parent	used	vmid
txz	65.96 MiB	local:vztmpl/centos-7-default_20171212_amd64.tar.xz			
txz	101.32 MiB	local:vztmpl/centos-8-default_20191016_amd64.tar.xz			

Terlihat *container template CentOS 8* berhasil diunggah ke *local storage* di server *PVE*.

e) Mengujicoba menghapus *container template CentOS 8* dengan mengeksekusi *playbook* "manajemen-container-template.yml" dan keseluruhan *task* didalamnya kecuali *task* dengan *tags unggah*.

```
# ansible-playbook manajemen-container-template.yml --skip-tags "unggah"
```

Hasil eksekusi dari *playbook* tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible-playbook manajemen-container-template.yml --skip-tags "unggah"
PLAY [Manajemen Container Template di PVE] *****
TASK [Gathering Facts] *****
ok: [pve]
TASK [include vars] *****
ok: [pve]
TASK [Hapus Container Template] *****
changed: [pve]
PLAY RECAP *****
pve                : ok=3   changed=1   unreachable=0   failed=0   skipped=0   rescued=
0   ignored=0
[root@ansible ~]#
```

- f) Memverifikasi *container template CentOS 8* yang telah dihapus menggunakan *ansible ad-hoc command*.

```
# ansible servers -m command -a "pvesh get /nodes/pve/storage/local/content"
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible servers -m command -a "pvesh get /nodes/pve/storage/local/content"
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at
/usr/bin/python, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.8/reference_appendices/interpreter_discovery.html for more
information.
pve | CHANGED | rc=0 >>
```

format	size	valid	parent	used	vmid
txz	65.96 MiB	local:vztmpl/centos-7-default_20171212_amd64.tar.xz			

Terlihat *container template CentOS 8* telah berhasil dihapus dari *local storage* di server *PVE*.

BAB XI

STUDI KASUS OTOMASI MANAJEMEN CONTAINER, USER DAN PERMISSION DARI PVE DENGAN ANSIBLE PLAYBOOK

Bab ini membahas tentang studi kasus otomasi menggunakan *ansible playbook* terkait objek pada *server PVE* yaitu manajemen *container* dan *user* serta *permission*. *Playbook* dari studi kasus yang dibahas pada bab ini dapat diunduh pada situs **Github** dengan alamat <https://github.com/iputuhariyadi/ansible-proxmox-playbook>.

A. MANAJEMEN CONTAINER

Adapun contoh penggunaan *ansible module proxmox* pada satu *file playbook* untuk memanajemen *multiple container* dan menggunakan *file* terpisah baik untuk menampung *variable* terkait akun otentikasi ke *server PVE* maupun *task* terkait pembuatan *instance* adalah sebagai berikut:

1. Membuat *file* *YAML* yang memuat *task* untuk pembuatan *container* dengan nama "container.yml".

```
# nano container.yml
```

Dengan konten sebagai berikut:

```
1 - name: Task Membuat Container Baru
2   proxmox:
3     api_host: "{{ api_host }}"
4     api_user: "{{ api_user }}"
5     api_password: "{{ api_password }}"
6     node: "{{ node }}"
7     vmid: "{{ item.vmid }}"
8     hostname: "{{ item.hostname }}"
9     password: "{{ item.ctpassword }}"
10    ostemplate: 'local:vztmpl/{{ ct_ostemplate }}'
11    cpus: 1
12    cpuunits: 500
13    memory: 512
14    swap: 512
```

```

15     disk: '5'
16     storage: local-lvm
17     nameserver: "{{ ct_nameserver }}"
18     searchdomain: "{{ ct_searchdomain }}"
19     netif: '{"net0": "name=eth0,gw={{ ct_gateway }},ip={{
        item.ip_address}},bridge=vibr0"}'
20     onboot: no
21     unprivileged: yes
22     state: present

```

Penjelasan:

- Baris 1: *name* digunakan untuk menentukan nama dari *task* yaitu “**Task Membuat Container Baru**”. Di dalam setiap *task* memuat kode yang berhubungan dengan *modul* yang harus dieksekusi.
- Baris 2: nama modul yang dieksekusi yaitu **proxmox**.
- Baris 3: merupakan parameter dari *ansible proxmox module* untuk menentukan *host* dari *Proxmox VE Cluster* yang nilainya diambil dari *variable* “`{{ api_host }}`”.
- Baris 4: merupakan parameter dari *ansible proxmox module* untuk menentukan *user* yang digunakan *login* ke *Proxmox VE* yang nilainya diambil dari *variable* “`{{ api_user }}`”.
- Baris 5: merupakan parameter dari *ansible proxmox module* untuk menentukan *password* otentikasi ke *Proxmox VE* yang nilainya diambil dari *variable* “`{{ api_password }}`”.
- Baris 6: merupakan parameter dari *ansible proxmox module* untuk menentukan *node* dari *Proxmox VE* sebagai lokasi *container template* yang nilainya diambil dari *variable* “`{{ node }}`”.
- Baris 7: digunakan untuk mengatur *ID instance* atau *container*.
- Baris 8: digunakan untuk mengatur *hostname* dari *instance* atau *container*.
- Baris 9: digunakan untuk mengatur sandi untuk *user root*.
- Baris 10: digunakan untuk mengatur *container template* yang digunakan untuk pembuatan *instance* atau *container*.

- Baris 11: digunakan untuk mengatur jumlah CPU yang dialokasikan untuk *instance* atau *container*.
- Baris 12: digunakan untuk mengatur *CPU weight* dari *container*, sebagai contoh 500.
- Baris 13: digunakan untuk mengatur ukuran memori dari *instance* dalam satuan *megabyte*, sebagai contoh **512**.
- Baris 14: digunakan untuk mengatur ukuran dari *swap* dalam satuan *megabyte*, sebagai contoh **512**.
- Baris 15: digunakan untuk mengatur ukuran dari *hardisk* dalam satuan *gigabyte*, sebagai contoh **5**.
- Baris 16: digunakan untuk mengatur media penyimpanan dari *container*, sebagai contoh **local-lvm**.
- Baris 17: digunakan untuk mengatur alamat IP dari *server DNS*.
- Baris 18: digunakan untuk nama domain pencarian untuk *container*.
- Baris 19: digunakan untuk mengatur *network interface* meliputi nama dari *interface*, alamat IP, *default gateway* dan *bridge*.
- Baris 20: digunakan untuk mengatur apakah *container* akan dijalankan (*started*) ketika *host* dari sistem melakukan *boot* yaitu berupa nilai *boolean* “*yes*” dan “*no*”. Sebagai contoh diatur **no**.
- Baris 21: digunakan untuk mengatur pemetaan semua *user ID (UID)* dan *group ID (GID)* pada *container* ke rentang nomor berbeda pada *host machine* dimana umumnya *root* memiliki *uid* 0 dipetakan menjadi *uid* 100000, *user id* 1 menjadi 100001 dan seterusnya. Nilai dari parameter ini berupa *boolean* yaitu *yes* dan *no*. Sebagai contoh diatur **yes**.
- Baris 22: digunakan untuk mengatur *state* dari *instance* yang meliputi *present* (membuat *container*), *started* (menjalankan), *absent* (menghapus *container*), *stopped* (menghentikan) dan *restarted* (menjalankan ulang atau *reboot*). Untuk membuat *instance* maka *state* yang digunakan adalah **present**.

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan tombol **Enter**.

Keluar dari *editor nano* dengan menekan tombol **CTRL+X**.

- Membuat *playbook* dengan "manajemen-multiple-container.yml" dan menyisipkan *file variable* "pve_auth_vars.yml" pada bagian *task* serta menyisipkan *file task* "container.yml" yang memuat parameter-parameter pembuatan *container*. Selain itu juga diterapkan atribut **tags** sehingga ketika *playbook* dieksekusi dapat memfilter *task* tertentu yang dijalankan yaitu dalam hal *task create, start, stop* atau **delete container**.

```
# nano manajemen-multiple-container.yml
```

Dengan konten sebagai berikut:

```
1 ---
2 - name: Manajemen Multiple Container di Proxmox
3   hosts: servers
4   vars:
5     ct_ostemplate: centos-7-default_20171212_amd64.tar.xz
6     ct_nameserver: 192.168.169.254
7     ct_searchdomain: "ubg.local"
8     ct_gateway: 192.168.169.254
9     containers:
10      - { vmid: 111, hostname: "server111.ubg.local",
11        ctpassword: "12345678", ip_address: 192.168.169.111/24 }
12      - { vmid: 112, hostname: "server112.ubg.local",
13        ctpassword: "12345678", ip_address: 192.168.169.112/24 }
14      - { vmid: 113, hostname: "server113.ubg.local",
15        ctpassword: "12345678", ip_address: 192.168.169.113/24 }
16 tasks:
17   - name: include vars
18     include_vars: pve_auth_vars.yml
19     no_log: true
20   - name: Membuat multiple container
21     include_tasks: container.yml
22     with_items: "{{ containers }}"
23     tags: create
24   - name: Menghentikan multiple container
25     proxmox:
26       api_host: "{{ api_host }}"
27       api_user: "{{ api_user }}"
28       api_password: "{{ api_password }}"
```

```

28     node: "{{ node }}"
29     vmid: "{{ item.vmid }}"
30     state: stopped
31 with_items: "{{ containers }}"
32 tags: stop
33
34 - name: Menjalankan multiple container
35   proxmox:
36     api_host: "{{ api_host }}"
37     api_user: "{{ api_user }}"
38     api_password: "{{ api_password }}"
39     node: "{{ node }}"
40     vmid: "{{ item.vmid }}"
41     state: started
42 with_items: "{{ containers }}"
43 tags: start
44
45 - name: Menghapus multiple container
46   proxmox:
47     api_host: "{{ api_host }}"
48     api_user: "{{ api_user }}"
49     api_password: "{{ api_password }}"
50     node: "{{ node }}"
51     vmid: "{{ item.vmid }}"
52     state: absent
53 with_items: "{{ containers }}"
54 tags: delete

```

Penjelasan:

- Baris 1: --- (3 *hyphen*) merupakan awal dari dokumen YAML.
- Baris 2: *name* digunakan untuk menentukan nama dari *ansible playbook* yaitu **“Manajemen Multiple Container di Proxmox”**.
- Baris 3: *hosts* digunakan untuk menentukan daftar *host* atau *group host* sebagai lokasi eksekusi *task* yaitu *group servers*.
- Baris 4: *vars* digunakan untuk mendefinisikan *variable* pada *playbook* dimana penggunaannya serupa dengan yang terdapat di bahasa pemrograman.
- Baris 5-8: deklarasi 4 (empat) *variable* yaitu **“ct_ostemplate”** dengan nilai **“centos-7-default_20171212_amd64.tar.xz”**, **“ct_nameserver”** dengan nilai **“192.168.169.254”**, **“ct_searchdomain”** dengan nilai **“ubg.local”** dan **“ct_gateway”** dengan nilai **“192.168.169.254”**.

- Baris 9-12: deklarasi variable bernama “**containers**” dengan struktur data *list of dictionaries* yang didalamnya memuat 3 (tiga) data untuk pembuatan container, seperti terlihat pada tabel berikut:

vmid	hostname	ctpassword	ip_address
111	server111.ubg.local	12345678	192.168.169.111/24
112	server112.ubg.local	12345678	192.168.169.112/24
113	server113.ubg.local	12345678	192.168.169.113/24

- Baris 13: *tasks* merupakan daftar tindakan atau aksi yang perlu dieksekusi.
- Baris 14: *name* digunakan untuk menentukan nama dari *task* yaitu “**include vars**”. Di dalam setiap *task* memuat kode yang berhubungan dengan *modul* yang harus dieksekusi.
- Baris 15: *include_vars* digunakan untuk memuat *variable* dari *file* yaitu “*pve_auth_vars.yml*”. *Variable* di dalam *file* tersebut diperlukan untuk pembentukan koneksi ke *server* PVE.
- Baris 16: digunakan agar tidak menampilkan hasil dari eksekusi *task* ketika *playbook* dijalankan menggunakan mode *verbose* atau *-v* sehingga menyembunyikan informasi sensitif yaitu *variable* terkait koneksi ke *server* PVE.
- Baris 18: *name* digunakan untuk menentukan nama dari *task* yaitu “**Membuat container template**”. Di dalam setiap *task* memuat kode yang berhubungan dengan *modul* yang harus dieksekusi.
- Baris 19: *include_tasks* digunakan untuk menyisipkan *file task* dengan nama “*container.yml*” yang didalamnya memuat instruksi untuk membuat *container* baru.
- Baris 20: digunakan untuk melakukan perulangan eksekusi *task* “**Membuat multiple container**” sejumlah nilai yang terdapat pada *variable* “*containers*” yaitu sebanyak 3 (tiga) kali sehingga akan terbentuk *container* masing-masing dengan **vmid 111, 112 dan 113**.
- Baris 21: *tags* untuk menandai *task* “**Membuat multiple container**” yaitu “**create**” sehingga ketika eksekusi *playbook* dapat memfilter *task* tertentu yang dijalankan.

- Baris 23: *name* digunakan untuk menentukan nama dari *task* yaitu “**Menghentikan multiple Container**”. Di dalam setiap *task* memuat kode yang berhubungan dengan *modul* yang harus dieksekusi.
- Baris 24: nama modul yang dieksekusi yaitu **proxmox**.
- Baris 25: merupakan parameter dari *ansible proxmox module* untuk menentukan *host* dari *Proxmox VE Cluster* yang nilainya diambil dari *variable* “`{{ api_host }}`”.
- Baris 26: merupakan parameter dari *ansible proxmox module* untuk menentukan *user* yang digunakan *login* ke *Proxmox VE* yang nilainya diambil dari *variable* “`{{ api_user }}`”.
- Baris 27: merupakan parameter dari *ansible proxmox module* untuk menentukan *password* otentikasi ke *Proxmox VE* yang nilainya diambil dari *variable* “`{{ api_password }}`”.
- Baris 28: merupakan parameter dari *ansible proxmox module* untuk menentukan *node* dari *Proxmox VE* sebagai lokasi *container template* yang nilainya diambil dari *variable* “`{{ node }}`”.
- Baris 29: merupakan parameter dari *ansible proxmox module* untuk menentukan *vmid* yang akan dihentikan dimana nilainya diambil dari *variable containers* menggunakan `{{ item.vmid }}`.
- Baris 30: digunakan untuk mengatur *state* dari *instance* yang meliputi *present* (membuat *container*), *started* (menjalankan), *absent* (menghapus *container*), *stopped* (menghentikan) dan *restarted* (menjalankan ulang atau *reboot*). Untuk menghentikan *instance* maka *state* yang digunakan adalah **absent**.
- Baris 31: digunakan untuk melakukan perulangan eksekusi *task* “**Menghentikan multiple container**” sejumlah nilai yang terdapat pada *variable* “*containers*” yaitu sebanyak 3 (tiga) kali. Dalam hal ini yang akan dihentikan adalah *container* dengan **ID 111, 112 dan 113**.

- Baris 32: *tags* untuk menandai *task* “**Menghentikan multiple container**” yaitu “**stop**” sehingga ketika eksekusi *playbook* dapat memfilter *task* tertentu yang dijalankan.
- Baris 34: *name* digunakan untuk menentukan nama dari *task* yaitu “**Menjalankan multiple Container**”. Di dalam setiap *task* memuat kode yang berhubungan dengan *modul* yang harus dieksekusi.
- Baris 35: nama modul yang dieksekusi yaitu **proxmox**.
- Baris 36: merupakan parameter dari *ansible proxmox module* untuk menentukan *host* dari *Proxmox VE Cluster* yang nilainya diambil dari *variable* “`{{ api_host }}`”.
- Baris 37: merupakan parameter dari *ansible proxmox module* untuk menentukan *user* yang digunakan *login* ke *Proxmox VE* yang nilainya diambil dari *variable* “`{{ api_user }}`”.
- Baris 38: merupakan parameter dari *ansible proxmox module* untuk menentukan *password* otentikasi ke *Proxmox VE* yang nilainya diambil dari *variable* “`{{ api_password }}`”.
- Baris 39: merupakan parameter dari *ansible proxmox module* untuk menentukan *node* dari *Proxmox VE* sebagai lokasi *container template* yang nilainya diambil dari *variable* “`{{ node }}`”.
- Baris 40: merupakan parameter dari *ansible proxmox module* untuk menentukan *vmid* yang akan dijalankan dimana nilainya diambil dari *variable containers* menggunakan `{{ item.vmid }}`.
- Baris 41: digunakan untuk mengatur *state* dari *instance* yang meliputi *present* (membuat *container*), *started* (menjalankan), *absent* (menghapus *container*), *stopped* (menghentikan) dan *restarted* (menjalankan ulang atau *reboot*). Untuk menjalankan *instance* maka *state* yang digunakan adalah **started**.
- Baris 42: digunakan untuk melakukan perulangan eksekusi *task* “**Menjalankan multiple container**” sejumlah nilai yang terdapat pada *variable* “*containers*” yaitu

sebanyak 3 (tiga) kali. Dalam hal ini yang akan dijalankan adalah *container* dengan **ID 111, 112 dan 113**.

- Baris 43: *tags* untuk menandai *task* “**Menjalankan multiple container**” yaitu “**start**” sehingga ketika eksekusi *playbook* dapat memfilter *task* tertentu yang dijalankan.
- Baris 45: *name* digunakan untuk menentukan nama dari *task* yaitu “**Menghapus multiple Container**”. Di dalam setiap *task* memuat kode yang berhubungan dengan *modul* yang harus dieksekusi.
- Baris 46: nama modul yang dieksekusi yaitu **proxmox**.
- Baris 47: merupakan parameter dari *ansible proxmox module* untuk menentukan *host* dari *Proxmox VE Cluster* yang nilainya diambil dari *variable* “`{{ api_host }}`”.
- Baris 48: merupakan parameter dari *ansible proxmox module* untuk menentukan *user* yang digunakan *login* ke *Proxmox VE* yang nilainya diambil dari *variable* “`{{ api_user }}`”.
- Baris 49: merupakan parameter dari *ansible proxmox module* untuk menentukan *password* otentikasi ke *Proxmox VE* yang nilainya diambil dari *variable* “`{{ api_password }}`”.
- Baris 50: merupakan parameter dari *ansible proxmox module* untuk menentukan *node* dari *Proxmox VE* sebagai lokasi *container template* yang nilainya diambil dari *variable* “`{{ node }}`”.
- Baris 51: merupakan parameter dari *ansible proxmox module* untuk menentukan *vmid* yang akan dihapus dimana nilainya diambil dari *variable* *containers* menggunakan `{{ item.vmid }}`.
- Baris 52: digunakan untuk mengatur *state* dari *instance* yang meliputi *present* (membuat *container*), *started* (menjalankan), *absent* (menghapus *container*), *stopped* (menghentikan) dan *restarted* (menjalankan ulang atau *reboot*). Untuk menjalankan *instance* maka *state* yang digunakan adalah **absent**.

- Baris 53: digunakan untuk melakukan perulangan eksekusi *task* “**Menghapus multiple container**” sejumlah nilai yang terdapat pada *variable* “*containers*” yaitu sebanyak 3 (tiga) kali. Dalam hal ini yang akan dihapus adalah *container* dengan ID **111, 112 dan 113**.
- Baris 54: *tags* untuk menandai *task* “**Menghapus multiple container**” yaitu “**delete**” sehingga ketika eksekusi *playbook* dapat memfilter *task* tertentu yang dijalankan.

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan tombol **Enter**.

Keluar dari *editor nano* dengan menekan tombol **CTRL+X**.

3. Mengujicoba pembuatan *multiple container* dengan mengeksekusi *playbook* “*manajemen-multiple-container.yml*” dan keseluruhan *task* kecuali **start**, **stop** dan **delete**.

```
# ansible-playbook manajemen-multiple-container.yml --skip-tags "start,stop,delete"
```

Hasil eksekusi dari *playbook* tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible-playbook manajemen-multiple-container.yml --skip-tags "start,stop,delete"
PLAY [Manajemen Multiple Container di Proxmox] *****
TASK [include vars] *****
ok: [pve]

TASK [Membuat multiple container] *****
included: /root/container.yml for pve
included: /root/container.yml for pve
included: /root/container.yml for pve

TASK [Task Membuat Container Baru] *****
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at /usr/bin/python, but
future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.8/reference_appendices/interpreter_discovery.html for more information.

changed: [pve]

TASK [Task Membuat Container Baru] *****
changed: [pve]

TASK [Task Membuat Container Baru] *****
changed: [pve]

PLAY RECAP *****
pve                : ok=7    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    igno
red=0
```

4. Memverifikasi hasil pembuatan *multiple container* menggunakan *ansible ad-hoc command*.

```
# ansible servers -m command -a "pvesh get /nodes/pve/lxc"
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible servers -m command -a "pvesh get /nodes/pve/lxc"
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at /usr/bin/python, but
future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.8/reference_appendices/interpreter_discovery.html for more information.

pve | CHANGED | rc=0 >>
+-----+-----+-----+-----+-----+-----+-----+-----+
| status | vmid  | cpus  | maxdisk | maxmem | maxswap | name                | uptime |
+-----+-----+-----+-----+-----+-----+-----+-----+
| stopped | 112   | 1     | 5.00 GiB | 512.00 MiB | 512.00 MiB | server112.ubg.local |         |
+-----+-----+-----+-----+-----+-----+-----+-----+
| stopped | 100   | 1     | 8.00 GiB | 1.00 GiB  | 512.00 MiB | server100.ubg.local |         |
+-----+-----+-----+-----+-----+-----+-----+-----+
| stopped | 111   | 1     | 5.00 GiB | 512.00 MiB | 512.00 MiB | server111.ubg.local |         |
+-----+-----+-----+-----+-----+-----+-----+-----+
| stopped | 113   | 1     | 5.00 GiB | 512.00 MiB | 512.00 MiB | server113.ubg.local |         |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

Terlihat 3 (tiga) *container* yaitu masing-masing dengan **vmid 111, 112 dan 113** berhasil dibuat.

5. Menguji coba menjalankan *multiple container* dengan mengeksekusi *playbook* "manajemen-multiple-container.yml" dan keseluruhan *task* kecuali **create, stop dan delete**.

```
# ansible-playbook manajemen-multiple-container.yml --skip-tags "create,stop,delete"
```

Hasil eksekusi dari *playbook* tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible-playbook manajemen-multiple-container.yml --skip-tags "create,stop,delete"

PLAY [Manajemen Multiple Container di Proxmox] *****

TASK [include vars] *****
ok: [pve]

TASK [Menjalankan multiple container] *****
changed: [pve] => (item={u'ctpassword': u'12345678', u'ip_address': u'192.168.169.111/24', u'hostname': u'server111.ubg.local', u'vmid': 111})
changed: [pve] => (item={u'ctpassword': u'12345678', u'ip_address': u'192.168.169.112/24', u'hostname': u'server112.ubg.local', u'vmid': 112})
changed: [pve] => (item={u'ctpassword': u'12345678', u'ip_address': u'192.168.169.113/24', u'hostname': u'server113.ubg.local', u'vmid': 113})
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at /usr/bin/python, but
future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.8/reference_appendices/interpreter_discovery.html for more information.

PLAY RECAP *****
pve                : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignore=0
red=0
```

6. Memverifikasi hasil menjalankan *multiple container* menggunakan *ansible ad-hoc command*.

```
# ansible servers -m command -a "pvesh get /nodes/pve/lxc"
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible servers -m command -a "pvesh get /nodes/pve/lxc"
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at /usr/bin/python, but
future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.8/reference_appendices/interpreter_discovery.html for more information.

pve | CHANGED | rc=0 >>
```

status	vmid	cpus	maxdisk	maxmem	maxswap	name	uptime
running	111	1	4.86 GiB	512.00 MiB	512.00 MiB	server111.ubg.local	2m 1s
running	112	1	4.86 GiB	512.00 MiB	512.00 MiB	server112.ubg.local	1m 57s
running	113	1	4.86 GiB	512.00 MiB	512.00 MiB	server113.ubg.local	1m 52s
stopped	100	1	8.00 GiB	1.00 GiB	512.00 MiB	server100.ubg.local	

Terlihat 3 (tiga) *container* yaitu masing-masing dengan **vmid 111, 112 dan 113** berhasil dijalankan dimana ditandai oleh **status running** dan nilai **uptime**.

- Mengujicoba menghentikan *multiple container* dengan mengeksekusi *playbook* "manajemen-multiple-container.yml" dan keseluruhan *task* kecuali **create, start dan delete**.

```
# ansible-playbook manajemen-multiple-container.yml --skip-tags "create,start,delete"
```

Hasil eksekusi dari *playbook* tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible-playbook manajemen-multiple-container.yml --skip-tags "create,start,delete"

PLAY [Manajemen Multiple Container di Proxmox] *****
TASK [include vars] *****
ok: [pve]

TASK [Menghentikan multiple container] *****
changed: [pve] => (item={u'ctpassword': u'12345678', u'ip_address': u'192.168.169.111/24', u'hostname': u'ser
ver111.ubg.local', u'vmid': 111})
changed: [pve] => (item={u'ctpassword': u'12345678', u'ip_address': u'192.168.169.112/24', u'hostname': u'ser
ver112.ubg.local', u'vmid': 112})
changed: [pve] => (item={u'ctpassword': u'12345678', u'ip_address': u'192.168.169.113/24', u'hostname': u'ser
ver113.ubg.local', u'vmid': 113})
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at /usr/bin/python, but
future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.8/reference_appendices/interpreter_discovery.html for more information.

PLAY RECAP *****
pve                : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    igno
red=0
```

- Memverifikasi hasil penghentian *multiple container* menggunakan *ansible ad-hoc command*.

```
# ansible servers -m command -a "pvesh get /nodes/pve/lxc"
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible servers -m command -a "pvesh get /nodes/pve/lxc"
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at /usr/bin/python, but
future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.8/reference_appendices/interpreter_discovery.html for more information.

pve | CHANGED | rc=0 >>
```

status	vmid	cpus	maxdisk	maxmem	maxswap	name	uptime
stopped	112	1	5.00 GiB	512.00 MiB	512.00 MiB	server112.ubg.local	
stopped	100	1	8.00 GiB	1.00 GiB	512.00 MiB	server100.ubg.local	
stopped	113	1	5.00 GiB	512.00 MiB	512.00 MiB	server113.ubg.local	
stopped	111	1	5.00 GiB	512.00 MiB	512.00 MiB	server111.ubg.local	

Terlihat 3 (tiga) *container* yaitu masing-masing dengan **vmid 111, 112 dan 113** berhasil dihentikan dimana ditandai dengan **status stopped**.

B. MANAJEMEN USER DAN PERMISSION

Adapun contoh penggunaan *ansible module command* pada satu *file playbook* untuk memajemen *user* dan *permission* terkait *container* dan menggunakan *file* terpisah untuk menampung adalah sebagai berikut:

1. Membuat *file* *YAML* yang memuat *variable* terkait data *user* dan *permission* yang akan dimanajemen baik proses pembuatan atau penghapusan dengan nama "user_permission_vars.yml".

```
# nano user_permission_vars.yml
```

Dengan konten sebagai berikut:

```
1 user_permissions:
2   - { userid: luna@pve, password: "12345678", firstname:
      "Luna", lastname: "Maya", comment: "Ilkom 2019", path:
      "/vms/111", role: "PVEVMUser" }
3   - { userid: ayu@pve, password: "12345678", firstname:
      "Ayu", lastname: "Tingting", comment: "RPL 2019", path:
      "/vms/112", role: "PVEVMAdmin" }
4   - { userid: via@pve, password: "12345678", firstname:
      "Via", lastname: "Vallen", comment: "TI 2019", path:
      "/vms/113", role: "PVEVMUser" }
```

Penjelasan:

Baris 1-4 merupakan deklarasi *variable* bernama "user_permissions" dengan struktur data *list of dictionaries* yang didalamnya memuat 3 (tiga) data untuk pembuatan user dan

pengaturan *permission* atau hak akses terhadap *container* di *server PVE*, seperti terlihat pada tabel berikut:

userid	password	firstname	lastname	comment	path	role
luna@pve	12345678	Luna	Maya	Ilkom 2019	/vms/111	PVEVMUser
ayu@pve	12345678	Ayu	Tingting	RPL 2019	/vms/112	PVEVMAdmin
via@pve	12345678	Via	Vallen	TI 2019	/vms/113	PVEVMUser

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan tombol **Enter**.

Keluar dari *editor nano* dengan menekan tombol **CTRL+X**.

2. Membuat *file* *YAML* yang memuat *task* untuk pembuatan *user* dan *permission* dengan nama "add-multiple-user-permission.yml".

```
# nano add-multiple-user-permission.yml
```

Dengan konten sebagai berikut:

```
1- name: Membuat User
2  command: pvesh create /access/users -userid "{{
  item.userid }}" -password "{{ item.password }}" -firstname
  "{{ item.firstname }}" -lastname "{{ item.lastname }}" -
  comment "{{ item.comment }}"
3
4- name: Membuat Permission
5  command: pvesh set /access/acl -path "{{ item.path }}" -
  users "{{ item.userid }}" -roles "{{ item.role }}"
```

Penjelasan:

- Baris 1: *name* digunakan untuk menentukan nama dari *task* yaitu "**Membuat user**". Di dalam setiap *task* memuat kode yang berhubungan dengan *modul* yang harus dieksekusi.
- Baris 2: nama modul yang dieksekusi yaitu **command**. Modul ini digunakan untuk mengeksekusi utilitas **pvesh** terkait pembuatan *user* baru di *server PVE*.
- Baris 4: *name* digunakan untuk menentukan nama dari *task* yaitu "**Membuat permission**". Di dalam setiap *task* memuat kode yang berhubungan dengan *modul* yang harus dieksekusi.

- Baris 5: nama modul yang dieksekusi yaitu **command**. Modul ini digunakan untuk mengeksekusi utilitas **pvesh** terkait pengaturan hak akses dari *user* tertentu di *server* PVE.

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan tombol **Enter**.

Keluar dari *editor nano* dengan menekan tombol **CTRL+X**.

3. Membuat *file* **YAML** yang memuat *task* untuk penghapusan *user* dan *permission* dengan nama "remove-multiple-user-permission.yml".

```
# nano remove-multiple-user-permission.yml
```

Dengan konten sebagai berikut:

```
1- name: Menghapus Permission
2  command: pvesh set /access/acl -delete -path "{{ item.path
   }}" -users "{{ item.userid }}" -roles "{{ item.role }}"
3
4- name: Menghapus User
5  command: pvesh delete /access/users/"{{ item.userid }}"
```

Penjelasan:

- Baris 1: *name* digunakan untuk menentukan nama dari *task* yaitu "**Menghapus Permission**". Di dalam setiap *task* memuat kode yang berhubungan dengan *modul* yang harus dieksekusi.
- Baris 2: nama modul yang dieksekusi yaitu **command**. Modul ini digunakan untuk mengeksekusi utilitas **pvesh** terkait penghapusan hak akses dari *user* tertentu di *server* PVE.
- Baris 4: *name* digunakan untuk menentukan nama dari *task* yaitu "**Menghapus User**". Di dalam setiap *task* memuat kode yang berhubungan dengan *modul* yang harus dieksekusi.
- Baris 5: nama modul yang dieksekusi yaitu **command**. Modul ini digunakan untuk mengeksekusi utilitas **pvesh** terkait penghapus *user* tertentu di *server* PVE.

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan tombol **Enter**.

Keluar dari *editor nano* dengan menekan tombol **CTRL+X**.

4. Membuat *playbook* dengan nama "manajemen-multiple-user-permission.yml" dan menyisipkan *file* *variable*

“user_permission_vars.yml”, “add-multiple-user-permission.yml” serta “remove-multiple-user-permission.yml” pada bagian *task*. Selain itu juga diterapkan atribut **tags** sehingga ketika *playbook* dieksekusi dapat mem-*filter task* tertentu yang dijalankan yaitu dalam hal *task* **add** atau **remove user permission** atau **reset password**.

```
# nano manajemen-multiple-user-permission.yml
```

Dengan konten sebagai berikut:

```
1 ---
2 - name: Manajemen User dan Permission di PVE
3   hosts: servers
4   gather_facts: false
5   vars:
6     new_password: "qwerty"
7   tasks:
8     - name: include vars
9       include_vars: user_permission_vars.yml
10
11    - name: Membuat User dan Permission
12      include_tasks: add-multiple-user-permission.yml
13      with_items: "{{ user_permissions }}"
14      tags: add
15
16    - name: Menghapus Permission dan User
17      include_tasks: remove-multiple-user-permission.yml
18      with_items: "{{ user_permissions }}"
19      tags: remove
20
21    - name: Reset Semua Password User
22      command: pvesh set /access/password --userid "{{
23        item.userid }}" --password "{{ new_password }}"
24      with_items: "{{ user_permissions }}"
25      tags: reset
```

Penjelasan:

- Baris 1: --- (3 *hyphen*) merupakan awal dari dokumen YAML.
- Baris 2: *name* digunakan untuk menentukan nama dari *ansible playbook* yaitu “Manajemen User dan Permission di PVE”.
- Baris 3: *hosts* digunakan untuk menentukan daftar *host* atau *group host* sebagai lokasi eksekusi *task* yaitu *group servers*.

- Baris 4: *gather_facts* digunakan untuk tidak melakukan pengambilan informasi terkait *remote host*.
- Baris 5: *vars* digunakan untuk mendefinisikan *variable* pada *playbook* dimana penggunaannya serupa dengan yang terdapat di bahasa pemrograman.
- Baris 6: deklarasi *variable* bernama “**new_password**” dengan nilai “**qwerty**” yang digunakan ketika ujicoba fitur *reset password user*.
- Baris 7: *tasks* merupakan daftar tindakan atau aksi yang perlu dieksekusi.
- Baris 8: *name* digunakan untuk menentukan nama dari *task* yaitu “**include vars**”. Di dalam setiap *task* memuat kode yang berhubungan dengan *modul* yang harus dieksekusi.
- Baris 9: *include_vars* digunakan untuk memuat *variable* dari *file* yaitu “*user_permission_vars.yml*”. *Variable* di dalam *file* tersebut memuat data terkait *user* dan *permission* yang dimanajemen pada *server PVE*.
- Baris 11: *name* digunakan untuk menentukan nama dari *task* yaitu “**Membuat User dan Permission**”. Di dalam setiap *task* memuat kode yang berhubungan dengan *modul* yang harus dieksekusi.
- Baris 12: *include_tasks* digunakan untuk menyisipkan file *task* dengan nama “*add-multiple-user-permission.yml*”. Di dalam memuat instruksi untuk membuat *user* baru dan mengatur hak akses dari setiap *user* baru yang ditambahkan dengan data yang berasal dari *variable* “*user_permissions*”.
- Baris 13: digunakan untuk melakukan perulangan eksekusi *task* “**Membuat User dan Permission**” sejumlah nilai yang terdapat pada *variable* “*user_permissions*” yaitu sebanyak 3 (tiga) kali sehingga akan terbentuk *user* masing-masing dengan **id luna, ayu** dan **via**. Selain itu setiap *user* tersebut diatur hak aksesnya sehingga **luna** dapat menggunakan **CTID 111** dengan *role* **PVEVMUser**, **ayu** dapat menggunakan **CTID 112** dengan *role* **PVEVMAdmin** dan **via** dapat menggunakan **CTID 113** dengan *role* **PVEVMUser**.
- Baris 14: *tags* untuk menandai *task* “**Membuat User dan Permission**” yaitu “**add**” sehingga ketika eksekusi *playbook* dapat memfilter *task* tertentu yang dijalankan.

- Baris 16: *name* digunakan untuk menentukan nama dari *task* yaitu **“Menghapus Permission dan User”**. Di dalam setiap *task* memuat kode yang berhubungan dengan *modul* yang harus dieksekusi.
- Baris 17: *include_tasks* digunakan untuk menyisipkan file *task* dengan nama **“remove-multiple-user-permission.yml”**. Di dalam memuat instruksi untuk menghapus hak akses dari *user* tertentu dan menghapus *user* pada server PVE berdasarkan data yang berasal dari *variable* **“user_permissions”**.
- Baris 18: digunakan untuk melakukan perulangan eksekusi *task* **“Menghapus Permission dan User”** sejumlah nilai yang terdapat pada *variable* **“user_permissions”** yaitu sebanyak 3 (tiga) kali sehingga akan menghapus ijin akses dari *user* masing-masing dengan **id luna** terhadap **CTITD 111**, **ayu** terhadap **CTITD 112** dan **via** terhadap **CTITD 113** serta menghapus ketida *user* tersebut dari *server* PVE.
- Baris 19: *tags* untuk menandai *task* **“Menghapus Permission dan User”** yaitu **“remove”** sehingga ketika eksekusi *playbook* dapat memfilter *task* tertentu yang dijalankan.
- Baris 21: *name* digunakan untuk menentukan nama dari *task* yaitu **“Reset Semua Password User”**. Di dalam setiap *task* memuat kode yang berhubungan dengan *modul* yang harus dieksekusi.
- Baris 22: nama *modul* yang dieksekusi yaitu **command**. *Modul* ini digunakan untuk mengeksekusi utilitas **pvesh** terkait perubahan sandi login dari *user* tertentu di *server* PVE.
- Baris 23: digunakan untuk melakukan perulangan eksekusi *task* **“Reset Semua Password User”** sejumlah nilai yang terdapat pada *variable* **“user_permissions”** yaitu sebanyak 3 (tiga) kali sehingga akan mengganti sandi login dari *user* dengan **id luna**, **ayu** dan **via** menggunakan nilai yang diambil dari *variable* **new_password** yaitu **“qwerty”**.

- Baris 24: *tags* untuk menandai *task* “Reset Semua Password User” yaitu “reset” sehingga ketika eksekusi *playbook* dapat memfilter *task* tertentu yang dijalankan.

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan tombol **Enter**.

Keluar dari *editor nano* dengan menekan tombol **CTRL+X**.

5. Mengujicoba pembuatan *multiple user permission* dengan mengeksekusi *playbook* “manajemen-multiple-user-permission.yml” dan keseluruhan *task* kecuali **remove** dan **reset**.

```
# ansible-playbook manajemen-multiple-user-permission.yml --skip-tags "remove,reset"
```

Hasil eksekusi *playbook* tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible-playbook manajemen-multiple-user-permission.yml --skip-tags "remove,reset"
PLAY [Manajemen User dan Permission di PVE] *****
TASK [include vars] *****
ok: [pve]

TASK [Membuat User dan Permission] *****
included: /root/add-multiple-user-permission.yml for pve
included: /root/add-multiple-user-permission.yml for pve
included: /root/add-multiple-user-permission.yml for pve

TASK [Membuat User] *****
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at /usr/bin/python, but
future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.8/reference_appendices/interpreter_discovery.html for more information.
changed: [pve]

TASK [Membuat Permission] *****
changed: [pve]

TASK [Membuat User] *****
changed: [pve]

TASK [Membuat Permission] *****
changed: [pve]

TASK [Membuat User] *****
changed: [pve]

TASK [Membuat Permission] *****
changed: [pve]

PLAY RECAP *****
pve          : ok=10  changed=6  unreachable=0  failed=0  skipped=0  rescued=0  igno
red=0

[root@ansible ~]#
```

6. Memverifikasi hasil pembuatan *multiple user* menggunakan *ansible ad-hoc command*.

```
# ansible servers -m command -a "pvesh get /access/users"
```

Hasil dari eksekusi perintah tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible servers -m command -a "pvesh get /access/users"
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at /usr/bin/python, but
future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.8/reference_appendices/interpreter_discovery.html for more information.

pve | CHANGED | rc=0 >>
```

userid	comment	email	enable	expire	firstname	keys	lastname
ali@pve	ali		1	0	ali		
ayu@pve	RPL 2019		1	0	Ayu		Tingting
hasan@pve	hasan		1	0	hasan		
luna@pve	Ilkom 2019		1	0	Luna		Maya
root@pam		putu.hariyadi@gmail.com	1	0			
via@pve	TI 2019		1	0	Via		Vallen

Terlihat *user luna, ayu dan via* berhasil dibuat.

- Memverifikasi hasil pembuatan *multiple user permission* menggunakan *ansible ad-hoc command*.

```
# ansible servers -m command -a "pvesh get /access/acl"
```

Hasil dari eksekusi perintah tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible servers -m command -a "pvesh get /access/acl"
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at /usr/bin/python, but
future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.8/reference_appendices/interpreter_discovery.html for more information.

pve | CHANGED | rc=0 >>
```

path	roleid	type	ugid	propagate
/vms/100	PVEVMUser	user	ali@pve	1
/vms/100	PVEVMAdmin	user	hasan@pve	1
/vms/111	PVEVMUser	user	luna@pve	1
/vms/112	PVEVMAdmin	user	ayu@pve	1
/vms/113	PVEVMUser	user	via@pve	1

Terlihat pengaturan *user permission* untuk *user luna, ayu dan via* berhasil dilakukan.

- Mengujicoba *reset* semua password user dengan mengeksekusi *playbook "manajemen-multiple-user-permission.yml"* dan keseluruhan *task* kecuali **add** dan **remove**.

```
# ansible-playbook manajemen-multiple-user-permission.yml --
skip-tags "add,remove"
```

Hasil eksekusi *playbook* tersebut, seperti terlihat pada gambar berikut:

```

[root@ansible ~]# ansible-playbook manajemen-multiple-user-permission.yml --skip-tags "add,remove"
PLAY [Manajemen User dan Permission di EVE] *****
TASK [include vars] *****
ok: [pve]

TASK [Reset Semua Password User] *****
changed: [pve] => (item={u'comment': u'Ilkom 2019', u'role': u'PVEVMUser', u'firstname': u'Luna', u'path': u'/vms/l11', u'lastname': u'Maya', u'password': u'12345678', u'userid': u'luna@pve'})
changed: [pve] => (item={u'comment': u'RPL 2019', u'role': u'PVEVMAdmin', u'firstname': u'Ayu', u'path': u'/vms/l12', u'lastname': u'Tingting', u'password': u'12345678', u'userid': u'ayu@pve'})
changed: [pve] => (item={u'comment': u'TI 2019', u'role': u'PVEVMUser', u'firstname': u'Via', u'path': u'/vms/l13', u'lastname': u'Vallen', u'password': u'12345678', u'userid': u'via@pve'})
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See https://docs.ansible.com/ansible/2.8/reference_appendices/interpreter_discovery.html for more information.

PLAY RECAP *****
pve                : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    igno
red=0

```

9. Lakukan verifikasi terkait hasil dari proses *reset password* dengan melakukan *login* menggunakan salah satu *user* meliputi “luna”, “ayu” atau “via” dengan *password* baru yaitu “qwerty” melalui *WebGUI Proxmox* yang diakses melalui alamat <https://192.168.169.1:8006>.
10. Mengujicoba penghapusan *multiple user permission* dengan mengeksekusi *playbook* “manajemen-multiple-user-permission.yml” dan keseluruhan *task* kecuali **add** dan **reset**.

```
# ansible-playbook manajemen-multiple-user-permission.yml --skip-tags "add,reset"
```

Hasil eksekusi *playbook* tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible-playbook manajemen-multiple-user-permission.yml --skip-tags "add,reset"
PLAY [Manajemen User dan Permission di PVE] *****
TASK [include vars] *****
ok: [pve]

TASK [Menghapus Permission dan User] *****
included: /root/remove-multiple-user-permission.yml for pve
included: /root/remove-multiple-user-permission.yml for pve
included: /root/remove-multiple-user-permission.yml for pve

TASK [Menghapus Permission] *****
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at /usr/bin/python, but
future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.8/reference_appendices/interpreter_discovery.html for more information.

changed: [pve]

TASK [Menghapus User] *****
changed: [pve]

TASK [Menghapus Permission] *****
changed: [pve]

TASK [Menghapus User] *****
changed: [pve]

TASK [Menghapus Permission] *****
changed: [pve]

TASK [Menghapus User] *****
changed: [pve]

PLAY RECAP *****
pve                : ok=10   changed=6   unreachable=0   failed=0   skipped=0   rescued=0   igno
red=0

[root@ansible ~]#
```

11. Memverifikasi hasil penghapusan *user permission* menggunakan *ansible ad-hoc command*.

```
# ansible servers -m command -a "pvesh get /access/acl"
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible servers -m command -a "pvesh get /access/acl"
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at /usr/bin/python, but
future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.8/reference_appendices/interpreter_discovery.html for more information.

pve | CHANGED | rc=0 >>
+-----+-----+-----+-----+-----+
| path   | roleid | type  | ugid  | propagate |
+-----+-----+-----+-----+-----+
| /vms/100 | PVEVMAdmin | user  | hasan@pve | 1          |
+-----+-----+-----+-----+-----+
| /vms/100 | PVEVMUser  | user  | ali@pve  | 1          |
+-----+-----+-----+-----+-----+
```

Terlihat *permission* untuk *user luna, ayu* dan *via* berhasil dihapus.

12. Memverifikasi hasil penghapusan *user* menggunakan *ansible ad-hoc command*.

```
# ansible servers -m command -a "pvesh get /access/users"
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible servers -m command -a "pvesh get /access/users"
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at /usr/bin/python, but
future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.8/reference_appendices/interpreter_discovery.html for more information.

pve | CHANGED | rc=0 >>
```

userid	comment	email	enable	expire	firstname	keys	lastname
ali@pve	ali		1	0	ali		
hasan@pve	hasan		1	0	hasan		
root@pam		putu.hariyadi@gmail.com	1	0			

Output menunjukkan bahwa user luna, ayu dan via telah berhasil dihapus.

- Mengujicoba menghapus *multiple container* dengan mengeksekusi *playbook* "manajemen-multiple-container.yml" dan keseluruhan *task* kecuali **create**, **start** dan **stop**.

```
# ansible-playbook manajemen-multiple-container.yml --skip-tags "create,start,stop"
```

Hasil dari eksekusi *playbook* tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible-playbook manajemen-multiple-container.yml --skip-tags "create,start,stop"

PLAY [Manajemen Multiple Container di Proxmox] *****

TASK [include vars] *****
ok: [pve]

TASK [Menghapus multiple container] *****
changed: [pve] => (item={u'ctpassword': u'12345678', u'ip_address': u'192.168.169.111/24', u'hostname': u'ser
ver111.ubg.local', u'vmid': 111})
changed: [pve] => (item={u'ctpassword': u'12345678', u'ip_address': u'192.168.169.112/24', u'hostname': u'ser
ver112.ubg.local', u'vmid': 112})
changed: [pve] => (item={u'ctpassword': u'12345678', u'ip_address': u'192.168.169.113/24', u'hostname': u'ser
ver113.ubg.local', u'vmid': 113})
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at /usr/bin/python, but
future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.8/reference_appendices/interpreter_discovery.html for more information.

PLAY RECAP *****
pve      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    igno
red=0
```

- Memverifikasi hasil penghapusan *multiple container* menggunakan *ansible ad-hoc command*.

```
# ansible servers -m command -a "pvesh get /nodes/pve/lxc"
```

Hasil eksekusi dari perintah tersebut, seperti terlihat pada gambar berikut:

```
[root@ansible ~]# ansible servers -m command -a "pvesh get /nodes/pve/lxc"
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at /usr/bin/python, but
future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.8/reference_appendices/interpreter_discovery.html for more information.

pve | CHANGED | rc=0 >>
```

status	vmid	cpus	maxdisk	maxmem	maxswap	name	uptime
stopped	100	1	8.00 GiB	1.00 GiB	512.00 MiB	server100.ubg.local	

Output menunjukkan bahwa **CT ID 111, 112 dan 113** telah berhasil dihapus.

DAFTAR REFERENSI

1. Ansible Documentation, 2019, <https://docs.ansible.com/>
2. Edureka, *What Is Ansible? – Configuration Management And Automation With Ansible*, 2019, <https://www.edureka.co/blog/what-is-ansible/>
3. Mikrotik, Mikrotik Documentation, 2019, https://wiki.mikrotik.com/wiki/Main_Page
4. Proxmox, Proxmox VE Administration Guide, 2019
5. Proxmox, Proxmox VE Wiki, 2019, https://pve.proxmox.com/wiki/Main_Page
6. Tutorialspoint, Ansible-Playbooks, 2019, https://www.tutorialspoint.com/ansible/ansible_playbooks.htm

TENTANG PENULIS



I Putu Hariyadi

adalah dosen di program studi Ilmu Komputer, [Universitas Bumigora](http://www.universitatumigora.ac.id), Mataram, Nusa Tenggara Barat (NTB). Penulis sangat antusias untuk mendalami dunia Teknologi Informasi & Komunikasi (TIK). Memiliki ketertarikan pada bidang Jaringan Komputer, *Network Programmability*, *Cloud Computing*, *Pemrograman Web* dan Keamanan Sistem Informasi serta Sistem Temu Kembali Informasi (*Information Retrieval*).

Sebagian besar pengalaman penulis ketika mengeksplorasi bidang tersebut dituangkan pada situs pribadi yang beralamat di <https://www.iputuhariyadi.net>. Untuk korespondensi dapat menghubungi penulis melalui email di alamat: admin@iputuhariyadi.net atau putu.hariyadi@universitasbumigora.ac.id.