

Laporan Implementasi Multi-Hypervisor Berbasis Proxmox dengan Integrasi KVM dan VirtualBox



DI SUSUN OLEH:

ANDRIAN YUZA SWANDA	(2401020157)
MUHAMMAD FAUZI	(2401020160)
MUHAMMAD KIKHA HAUKAL	(2401020140)
SAIF ALIF ABYAN	(2401020163)

FAKULTAS TEKNIK DAN TEKNOLOGI KEMARITIMAN

UNIVERSITAS MARITIM RAJA ALI HAJI

2025

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan penyertaan-Nya sehingga laporan mengenai Implementasi Multi-Hypervisor Berbasis Proxmox dengan Integrasi KVM dan VirtualBox ini dapat diselesaikan dengan baik. Laporan ini disusun sebagai bentuk dokumentasi teknis serta analisis terhadap fleksibilitas infrastruktur virtualisasi dalam mengelola berbagai lingkungan mesin virtual pada satu platform yang terpusat.

Fokus utama dari laporan ini adalah mengeksplorasi kemampuan Proxmox VE dalam menjalankan KVM sebagai hypervisor utama, sekaligus mengintegrasikan VirtualBox untuk kebutuhan spesifik yang memerlukan kompatibilitas perangkat lunak tertentu. Penulis berharap dokumentasi ini dapat memberikan gambaran yang jelas mengenai langkah-langkah konfigurasi, optimalisasi sumber daya, serta solusi atas tantangan teknis yang muncul selama proses implementasi berlangsung.

Terima kasih disampaikan kepada semua pihak yang telah memberikan bimbingan, dukungan, serta ilmu pengetahuan sehingga proyek ini dapat berjalan dengan lancar. Penulis menyadari bahwa masih terdapat ruang untuk pengembangan lebih lanjut dalam laporan ini. Oleh karena itu, segala kritik dan saran yang membangun akan sangat berharga untuk penyempurnaan di masa yang akan datang. Semoga laporan ini dapat memberikan manfaat dan wawasan bagi para pembaca, terutama yang bergelut di bidang administrasi sistem dan infrastruktur jaringan.

DAFTAR ISI

DAFTAR ISI.....	3
BAB I PENDAHULUAN.....	4
A. Abstrak.....	4
B. Latar Belakang	5
C. Rumusan Masalah.....	6
D. Tujuan.....	6
BAB II LANDASAN TEORI	7
A. Virtualisasi dan Hypervisor	7
B. Proxmox VE	7
C. Interopabilitas.....	9
BAB III PERANCANGAN DAN IMPLEMENTASI	10
A. Spesifikasi.....	10
B. Instalasi Hypervisor (Proxmox VE).....	10
C. Konfigurasi Integrasi	11
D. Konfigurasi Jaringan	12
BAB IV IMPLEMENTASI	13
A. Backup Snapshot.....	13
BAB V PENGUJIAN & ANALISIS	16
BAB VI KESIMPULAN.....	18
DAFTAR PUSTAKA	19
LAMPIRAN.....	20
A. Screenshot Setelah Konfigurasi Ubuntu Live Server	20
B. Summary Peforma dari Website Proxmox	20
C. Bukti Kerja	21

BAB I

PENDAHULUAN

A. Abstrak

Virtualisasi merupakan teknologi yang berperan penting dalam pengelolaan infrastruktur teknologi informasi modern karena mampu mengoptimalkan pemanfaatan sumber daya perangkat keras. Dalam praktiknya, sering kali dibutuhkan lebih dari satu jenis hypervisor untuk memenuhi kebutuhan pengelolaan sistem yang beragam. Oleh karena itu, penelitian ini berfokus pada implementasi lingkungan multi-hypervisor dengan memanfaatkan Proxmox Virtual Environment (VE) berbasis KVM serta Oracle VM VirtualBox pada sistem operasi Ubuntu Live Server.

Tujuan dari proyek ini adalah untuk memahami interoperabilitas antar-hypervisor, menguji kemampuan integrasi jaringan, serta menganalisis fitur snapshot, backup, dan performa sistem yang dihasilkan. Metode yang digunakan meliputi instalasi Proxmox VE sebagai hypervisor utama, konfigurasi mesin virtual, pengujian snapshot dan rollback, serta pemantauan performa menggunakan utilitas monitoring sistem.

Hasil implementasi menunjukkan bahwa integrasi multi-hypervisor dapat berjalan dengan baik tanpa menggunakan konsep nested virtualization. Selain itu, fitur snapshot dan backup pada Proxmox VE terbukti efektif sebagai mekanisme pengamanan data dan tidak memberikan dampak signifikan terhadap performa mesin virtual. Dengan demikian, implementasi multi-hypervisor ini dapat menjadi solusi yang fleksibel dan efisien untuk kebutuhan pembelajaran maupun pengelolaan infrastruktur virtual berskala kecil hingga menengah.

Kata kunci: Virtualisasi, Multi-Hypervisor, Proxmox VE, KVM, VirtualBox, Backup, Snapshot.

B. Latar Belakang

Virtualisasi telah menjadi standar dalam pengelolaan infrastruktur IT untuk mengoptimalkan penggunaan sumber daya perangkat keras. Dalam implementasi praktis, sering kali dibutuhkan lebih dari satu jenis hypervisor dalam satu sistem untuk menangani kebutuhan yang berbeda. Proxmox VE yang memanfaatkan teknologi KVM/QEMU unggul sebagai platform virtualisasi terpusat untuk mengelola mesin virtual dan container melalui antarmuka web, termasuk fitur manajemen storage, jaringan, snapshot, dan backup. Sementara Oracle VM VirtualBox sering digunakan karena kemudahannya dalam manajemen mesin virtual berbasis grafis, cocok untuk kebutuhan pembelajaran/desktop, serta dukungan format VM yang luas.

Laporan ini membahas mengenai implementasi multi-hypervisor pada sistem operasi Ubuntu Live Server. Dalam skenario ini, Ubuntu berfungsi sebagai Host OS tunggal yang mengintegrasikan dua teknologi hypervisor sekaligus, yaitu KVM dan VirtualBox. Berbeda dengan metode nested, pendekatan ini menempatkan kedua hypervisor langsung di atas kernel Ubuntu yang sama. Hal ini menuntut manajemen sumber daya yang ketat agar tidak terjadi konflik perebutan instruksi virtualisasi perangkat keras (VT-x atau AMD-V) antar kedua platform tersebut.

Implementasi ini mencakup seluruh siklus hidup mesin virtual, mulai dari proses instalasi, konfigurasi jaringan agar antar-VM dapat saling terhubung, hingga pengujian fitur krusial seperti snapshot dan backup. Melalui proyek ini, akan dianalisis bagaimana interoperabilitas antar-hypervisor dapat berjalan stabil dan bagaimana dampak performa yang dihasilkan pada lingkungan produksi yang menggunakan lebih dari satu penyedia virtualisasi.

C. Rumusan Masalah

1. Bagaimana cara mengintegrasikan KVM dan VirtualBox secara bersamaan di atas sistem operasi Ubuntu Live Server?
2. Bagaimana mengonfigurasi jaringan virtual agar mesin virtual dari kedua hypervisor yang berbeda dapat saling berkomunikasi?
3. Sejauh mana fitur snapshot dan backup pada masing-masing hypervisor menjamin keamanan data?
4. Bagaimana karakteristik performa sistem saat menjalankan beban kerja dari dua hypervisor berbeda secara simultan?

D. Tujuan

1. Membangun infrastruktur server yang mampu menjalankan teknologi KVM dan VirtualBox secara berdampingan pada Ubuntu Live Server.
2. Memahami prosedur integrasi dan konfigurasi teknis untuk mencapai interoperabilitas antar-hypervisor.
3. Melakukan pengujian fungsionalitas snapshot dan backup sebagai bagian dari manajemen risiko data.
4. Menganalisis efisiensi penggunaan sumber daya perangkat keras melalui laporan performa yang dihasilkan.

BAB II

LANDASAN TEORI

A. Virtualisasi dan Hypervisor

Virtualisasi merupakan teknologi yang memungkinkan satu perangkat keras fisik menjalankan beberapa sistem operasi secara bersamaan dengan cara membagi sumber daya seperti CPU, RAM, dan storage secara virtual. Teknologi ini memungkinkan efisiensi penggunaan sumber daya serta mempermudah pengelolaan sistem.

Rahmadi Rakhman, Nur Ayu Khalifah , Wahyudi Antoni (2024) Hypervisor dapat digunakan untuk “mengelola beberapa VM di satu komputer, memastikan setiap VM mendapatkan sumber daya yang dialokasikan. Hypervisor terbagi menjadi dua tipe: Type-1 (bare-metal) dan Type-2 (hosted)”. (Halaman 81)

seperti KVM yang digunakan pada Proxmox VE, berjalan langsung di atas perangkat keras tanpa sistem operasi host, sehingga memiliki performa dan keamanan yang lebih baik. Sedangkan hypervisor Type-2, seperti VirtualBox, berjalan di atas sistem operasi host, sehingga lebih mudah digunakan namun memiliki performa yang relatif lebih rendah.

(Rahmadi Rakhman, Nur Ayu Khalifah , Wahyudi Antoni 2024)

B. Proxmox VE

Setelah Bobi Agustian , Dodi Susanto (2018) menunjukkan Proxmox “merupakan salah satu software virtualisasi yang berbasis open source. Yang berarti proxmox ini bersifat gratis. Proxmox berbasis debian x86_64, oleh karenanya Proxmox hanya bisa diinstall pada mesin berbasis 64 bit, Proxmox tidak akan bisa diinstall pada mesin berbasis i386. Proxmox support beberapa jenis platform virtualisasi seperti KVM dan OpenVZ. Untuk menggunakan KVM pada proxmox anda juga harus memastikan processor yang anda gunakan support fitur virtualisasi pada hardware (hardware virtualization) baik intel VT atau AMD-V, berbeda dengan KVM, jika anda menggunakan OpenVZ anda tidak membutuhkan hardware virtualization ini.” (Halaman 135)

Fitur utama Proxmox VE meliputi manajemen mesin virtual, pengaturan jaringan virtual, manajemen storage, backup dan restore, serta monitoring sistem secara real-time. Dengan fitur tersebut, Proxmox VE banyak digunakan sebagai solusi virtualisasi pada server dan lingkungan laboratorium. (Bobi Agustian , Dodi Susanto 2018)

C. Interoperabilitas

Interoperabilitas adalah kemampuan dua atau lebih sistem yang berbeda untuk saling berkomunikasi, bertukar data, dan bekerja bersama secara efektif. Dalam lingkungan virtualisasi, interoperabilitas memungkinkan mesin virtual dengan sistem operasi atau konfigurasi yang berbeda untuk tetap dapat saling terhubung melalui jaringan virtual. Interoperabilitas sangat penting dalam sistem terdistribusi karena memungkinkan integrasi antar sistem tanpa harus bergantung pada satu platform tertentu. Dengan adanya interoperabilitas, pengelolaan dan pertukaran data dapat dilakukan secara lebih fleksibel dan efisien.

BAB III

PERANCANGAN DAN IMPLEMENTASI

A. Spesifikasi

Spesifikasi Device:

Device: Laptop Advan Heritage

RAM/ROM: 16 GB / 512 GB

CPU: Ryzen 5 7435 HS

OS: Windows 11

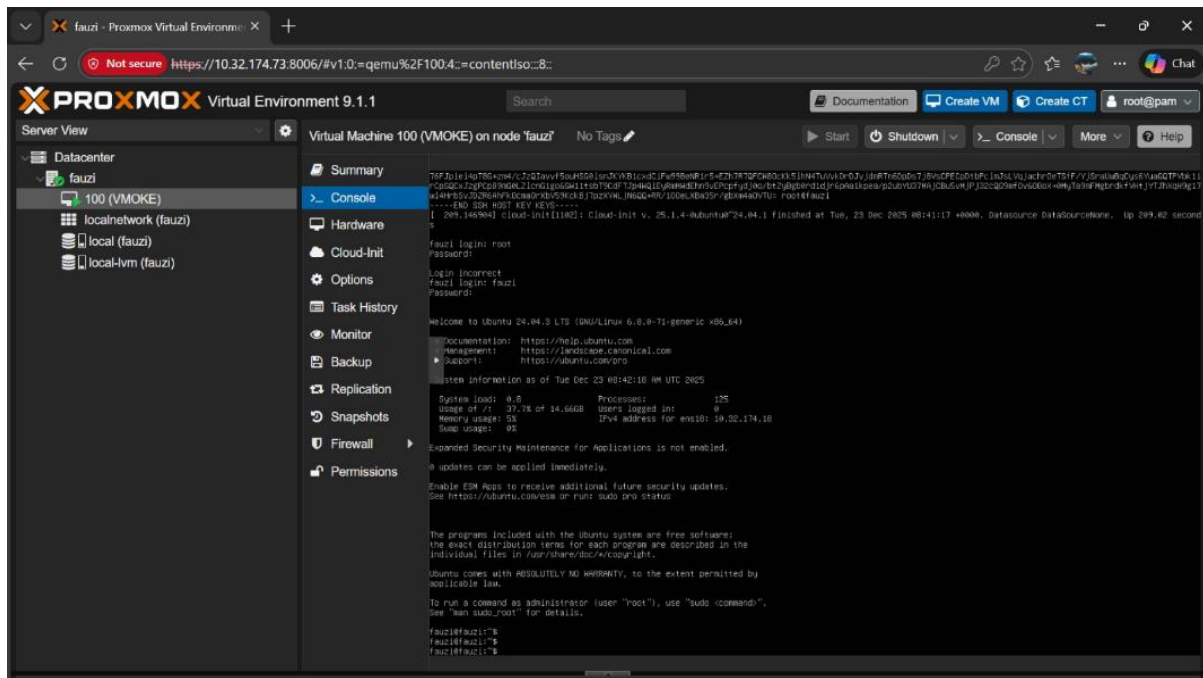
Spesifikasi lingkungan kerja:

1. Proxmox VE 9.1-1
2. Ubuntu Live Server 24.04.03 LS
3. Oracle Virtual Box

B. Instalasi Hypervisor (Proxmox VE)

Tahap instalasi hypervisor dilakukan dengan menggunakan Proxmox Virtual Environment (Proxmox VE) yang diinstal secara langsung pada perangkat keras (bare-metal). Proses ini diawali dengan pembuatan media instalasi menggunakan berkas ISO Proxmox VE, kemudian sistem dijalankan melalui media tersebut untuk memulai instalasi.

Selama proses instalasi, dilakukan konfigurasi dasar sistem seperti pemilihan media penyimpanan, pengaturan zona waktu, serta penentuan alamat IP statis untuk manajemen Proxmox VE. Setelah instalasi selesai, sistem melakukan reboot dan Proxmox VE dapat diakses melalui antarmuka berbasis web menggunakan browser.



Selanjutnya dilakukan verifikasi terhadap modul Kernel-based Virtual Machine (KVM) untuk memastikan dukungan virtualisasi aktif dan berfungsi dengan baik. Instalasi hypervisor ini dijalankan tanpa menggunakan konsep nested virtualization, karena Proxmox VE beroperasi langsung di atas perangkat keras dan tidak berada di dalam lingkungan virtual lainnya.

C. Konfigurasi Integrasi

Tahap konfigurasi integrasi menjelaskan pengaturan penggunaan dua platform virtualisasi, yaitu Proxmox VE (KVM) dan VirtualBox, yang dijalankan pada lingkungan yang terpisah. VirtualBox dijalankan langsung pada sistem operasi host, sedangkan Proxmox VE berfungsi sebagai hypervisor bare-metal. Dengan demikian, tidak diterapkan konsep nested virtualization, karena tidak terdapat hypervisor yang berjalan di dalam mesin virtual.

Integrasi antara kedua lingkungan virtualisasi ini tidak dilakukan melalui penanaman satu hypervisor ke dalam hypervisor lainnya, melainkan melalui penyelarasan konfigurasi jaringan dan pengujian konektivitas antar mesin virtual. Pendekatan ini memungkinkan kedua platform tetap dapat digunakan dalam satu skenario pengujian tanpa saling bergantung secara langsung.

D. Konfigurasi Jaringan

```
fauzi@fauzi:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether bc:24:11:06:56:45 brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 10.32.174.18/24 metric 100 brd 10.32.174.255 scope global dynamic ens18
        valid_lft 3172sec preferred_lft 3172sec
    inet6 fe80::be24:11ff:fe06:5645/64 scope link
        valid_lft forever preferred_lft forever
fauzi@fauzi:~$ ip route
default via 10.32.174.254 dev ens18 proto dhcp src 10.32.174.18 metric 100
10.32.174.0/24 dev ens18 proto kernel scope link src 10.32.174.18 metric 100
10.32.174.254 dev ens18 proto dhcp scope link src 10.32.174.18 metric 100
fauzi@fauzi:~$
```

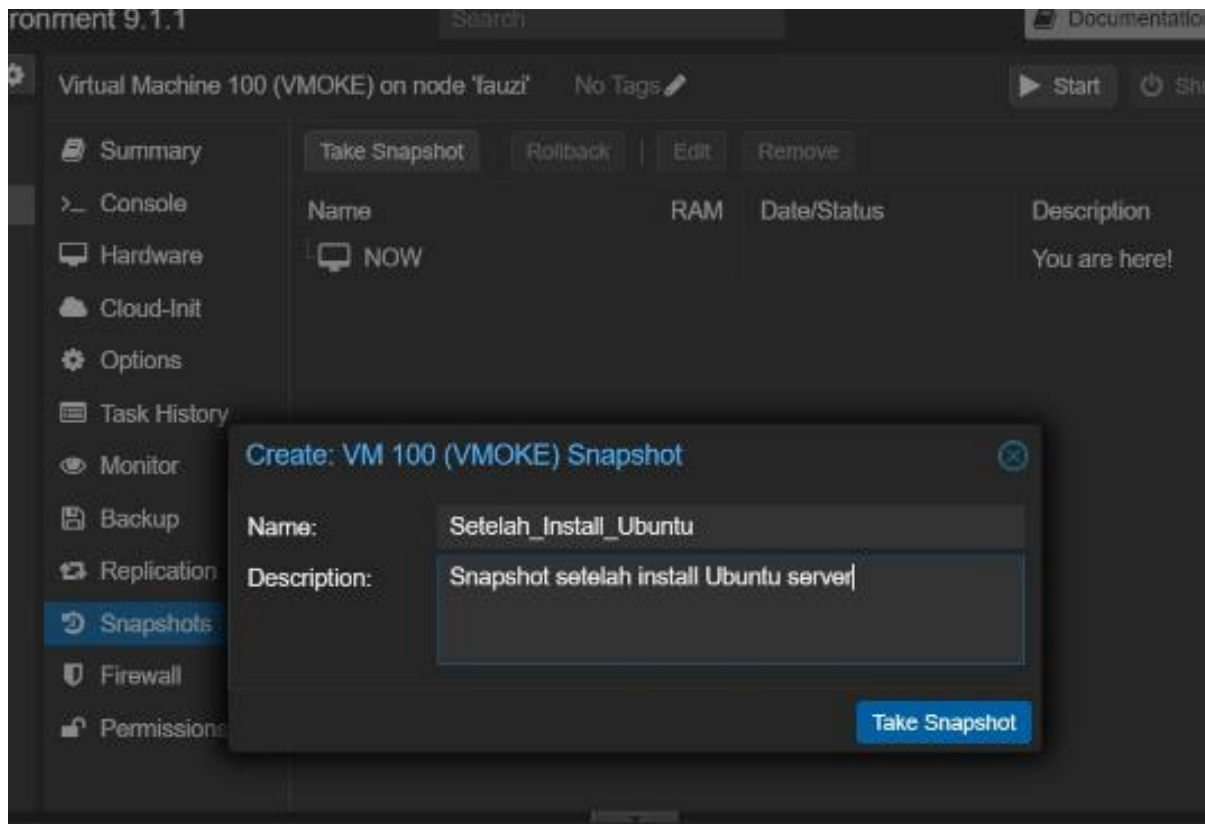
Konfigurasi jaringan dilakukan untuk memungkinkan komunikasi antara mesin virtual yang berjalan pada Proxmox VE (KVM) dan VirtualBox. Pada Proxmox VE, digunakan mekanisme virtual bridge yang menghubungkan mesin virtual ke jaringan fisik host sehingga VM memperoleh alamat IP dalam satu segmen jaringan yang sama.

Sementara itu, pada VirtualBox, pengaturan jaringan VM dikonfigurasi menggunakan mode Bridged Adapter, sehingga mesin virtual dapat terhubung langsung ke jaringan yang sama dengan Proxmox VE. Dengan konfigurasi ini, mesin virtual dari kedua platform dapat saling berkomunikasi tanpa memerlukan nested virtualization.

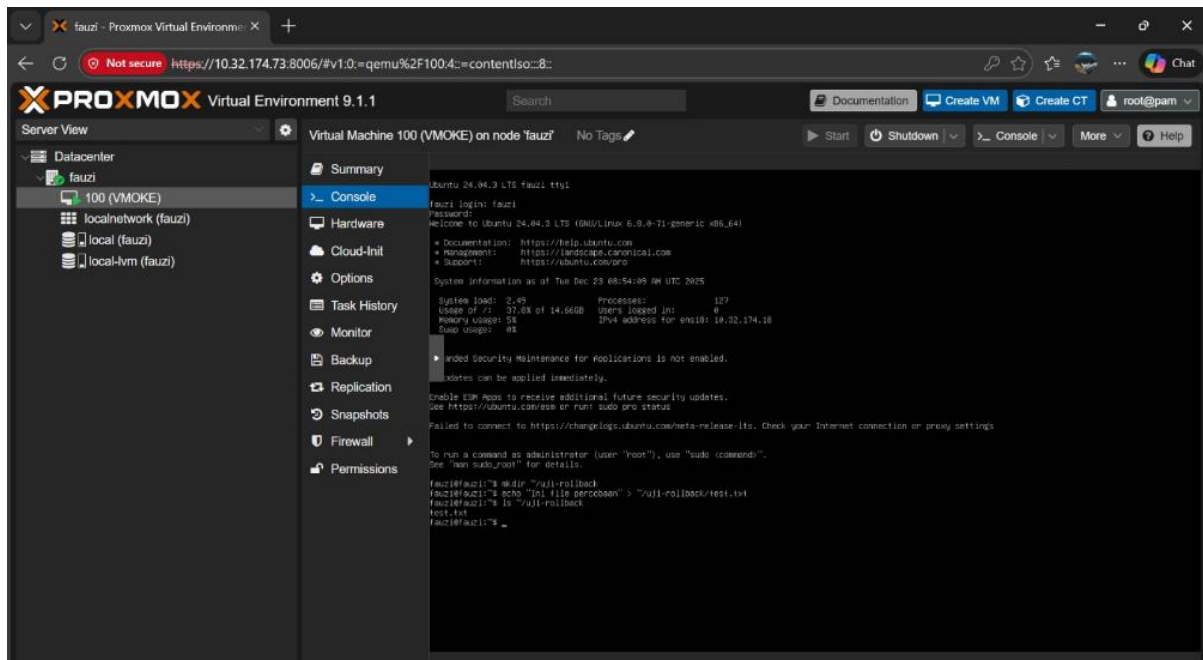
Keberhasilan konfigurasi jaringan diuji dengan melakukan pengujian konektivitas berupa perintah ping antar mesin virtual serta percobaan transfer data, yang menunjukkan bahwa komunikasi jaringan berjalan dengan baik.

BAB IV IMPLEMENTASI

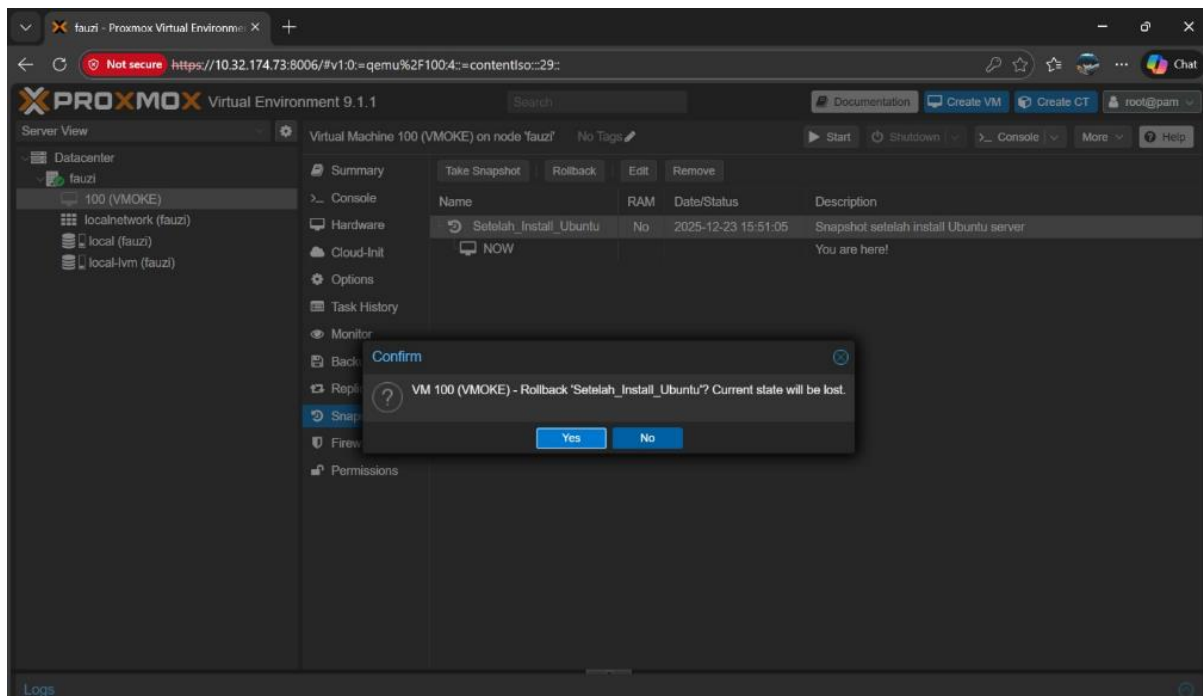
A. Backup Snapshot

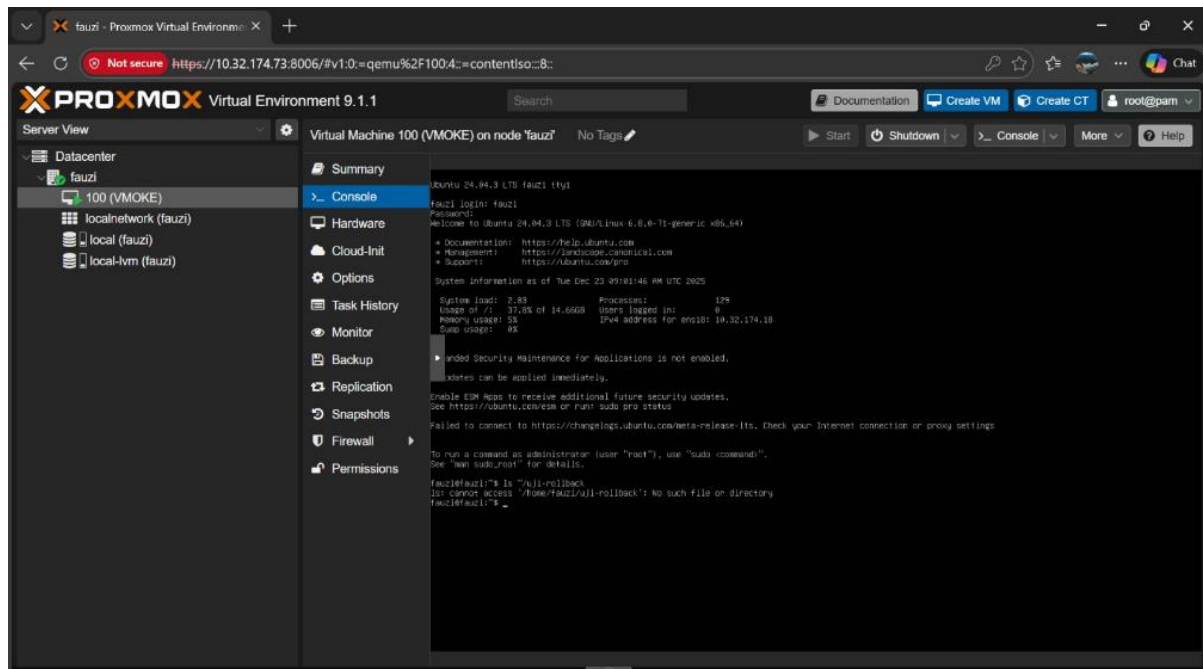


Implementasi backup snapshot dilakukan pada mesin virtual di lingkungan Proxmox VE dengan memanfaatkan fitur snapshot bawaan. Snapshot dibuat sebagai titik awal untuk pengujian mekanisme rollback tanpa melakukan perubahan konfigurasi sistem.



Setelah snapshot berhasil dibuat, dilakukan pengujian dengan menambahkan data baru pada mesin virtual, yaitu membuat sebuah folder bernama uji rollback dan sebuah file test.txt di dalam folder tersebut. Penambahan data ini bertujuan untuk mensimulasikan perubahan pada sistem setelah snapshot dibuat.





Selanjutnya, mesin virtual dikembalikan ke kondisi snapshot yang telah dibuat sebelumnya. Hasil implementasi menunjukkan bahwa folder uji rollback beserta file test.txt tidak lagi tersedia setelah proses pemulihan, yang menandakan bahwa snapshot berhasil berfungsi sebagai mekanisme backup dan rollback.

PENGUJIAN & ANALISIS

Laporan performa mesin virtual dilakukan sebagai kelanjutan dari implementasi backup snapshot yang telah dijelaskan pada poin sebelumnya. Pemantauan ini bertujuan untuk mengetahui kondisi penggunaan sumber daya sistem setelah proses pembuatan snapshot serta pengujian rollback dilakukan.

```

01                                     0.0% Tasks: 20, 22 thr, 195 khr; 1 running
11                                     0.5% Load average: 9.29 0.59 0.39
21                                     0.5% Uptime: 00:04:23
311                                  2.0%
Mem[|||||] 172M/3.82G
Swap[ ] 0K/3.86G

Main [Z]
PID USER PRI NI VIRT RES SHR S CPU# MEM% TIME+ Command
1 root 20 0 270440 17700 9244 S 0.5 0.3 0:04.66 /sbin/init
399 root RT 0 2820 27000 8440 S 0.5 0.7 0:00.45 /sbin/multiioind -d -s
594 systemd- 20 0 91024 7000 6912 S 0.5 0.2 0:01.48 /usr/lib/systemd/systemd-timedsyncd
670 systemd- 20 0 91024 7000 6912 S 0.5 0.2 0:00.20 /usr/lib/systemd/systemd-timedsyncd
1037 fauzi 20 0 8712 4952 3712 S 0.6 0.1 0:00.84 http
331 root 19 1 50444 16000 15704 S 0.0 0.4 0:00.67 /usr/lib/systemd/systemd-journald
332 root RT 0 2820 27000 8440 S 0.0 0.7 0:01.05 /sbin/multiioind -d -s
394 root 20 0 2820 27000 8440 S 0.0 0.7 0:00.03 /sbin/multiioind -d -s
396 root 20 0 2820 27000 8440 S 0.0 0.7 0:00.03 /sbin/multiioind -d -s
397 root RT 0 2820 27000 8440 S 0.0 0.7 0:00.02 /sbin/multiioind -d -s
398 root RT 0 2820 27000 8440 S 0.0 0.7 0:00.04 /sbin/multiioind -d -s
400 root RT 0 2820 27000 8440 S 0.0 0.7 0:00.02 /sbin/multiioind -d -s
481 root 20 0 25172 7552 4064 S 0.0 0.2 0:00.32 /usr/lib/systemd/systemd-udev
572 systemd- 20 0 10000 3000 8064 S 0.0 0.2 0:02.17 /usr/lib/systemd/systemd-networkd
587 systemd- 20 0 21504 15672 10436 S 0.0 0.3 0:05.29 /usr/lib/systemd/systemd-resolved
702 systemd- 20 0 9012 5120 4000 S 0.0 0.1 0:02.15 edbus-daemon --system --addresssystemd: --nofork --nopidfile --system-activation --syslog-o
706 polkitd 20 0 3740 3472 7296 S 0.0 0.2 0:02.61 /usr/lib/polkit-1/polkitd --no-debug
713 root 20 0 17904 8440 7552 S 0.0 0.2 0:01.72 /usr/lib/systemd/systemd-logind
710 root 20 0 457M 13104 11264 S 0.0 0.3 0:02.01 /usr/libexec/udisks2/udisksd
734 root 20 0 457M 13104 11264 S 0.0 0.3 0:00.00 /usr/libexec/udisks2/udisksd
735 root 20 0 457M 13104 11264 S 0.0 0.3 0:00.00 /usr/libexec/udisks2/udisksd
746 root 20 0 457M 13104 11264 S 0.0 0.3 0:00.00 /usr/libexec/udisks2/udisksd
748 root 20 0 4824 8016 2680 S 0.0 0.1 0:00.27 /usr/sbin/cron -f -P
752 root 20 0 107M 25250 13312 S 0.0 0.6 0:04.15 /usr/bin/python /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
753 syslog 20 0 217M 5760 4352 S 0.0 0.1 0:00.97 /usr/sbin/rsyslogd -n -iNONE
756 polkitd 20 0 3740 3472 7296 S 0.0 0.2 0:00.12 /usr/lib/polkit-1/polkitd --no-debug
757 polkitd 20 0 3740 3472 7296 S 0.0 0.2 0:00.00 /usr/lib/polkit-1/polkitd --no-debug
758 polkitd 20 0 3740 3472 7296 S 0.0 0.2 0:00.01 /usr/lib/polkit-1/polkitd --no-debug
760 root 20 0 457M 13104 11264 S 0.0 0.3 0:00.31 /usr/sbin/ModemManager
797 root 20 0 457M 13104 11264 S 0.0 0.3 0:00.00 /usr/libexec/udisks2/udisksd
813 syslog 20 0 217M 5760 4352 S 0.0 0.1 0:00.31 /usr/sbin/rsyslogd -n -iNONE
814 syslog 20 0 217M 5760 4352 S 0.0 0.1 0:00.06 /usr/sbin/rsyslogd -n -iNONE
816 syslog 20 0 217M 5760 4352 S 0.0 0.1 0:00.15 /usr/sbin/rsyslogd -n -iNONE
820 root 20 0 457M 13104 11264 S 0.0 0.3 0:00.00 /usr/libexec/udisks2/udisksd
821 root 20 0 6940 4400 3040 S 0.0 0.1 0:01.42 /bin/login -p --
832 root 20 0 2820 13416 10624 S 0.0 0.3 0:00.00 /usr/sbin/ModemManager
833 root 20 0 2820 13416 10624 S 0.0 0.3 0:00.00 /usr/sbin/ModemManager
835 root 20 0 2820 13416 10624 S 0.0 0.3 0:00.00 /usr/sbin/ModemManager
870 root 20 0 107M 25250 13312 S 0.0 0.6 0:00.00 /usr/bin/python /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
help [Setup Search] [litter tree] [lsntr] [nice -] [nice -] [kill] [logout]

```

The screenshot displays the Proxmox Virtual Environment (VE) 9.1.1 web interface. The left sidebar shows the 'Server View' with a tree structure under 'Datacenter' containing a cluster named 'fauzi'. Inside 'fauzi', there are several virtual machines listed: '100 (VMOKE)', 'localnetwork (fauzi)', 'local (fauzi)', and 'local-lvm (fauzi)'. The '100 (VMOKE)' VM is selected, and its configuration page is displayed. The page title is 'Virtual Machine 100 (VMOKE) on node 'fauzi''. The main content area shows the 'Summary' tab, which includes a table of VM statistics and a 'Notes' section. The statistics table shows the VM is running, has no HA state, and is on node 'fauzi'. It also shows CPU usage at 1.50%, memory usage at 9.85%, host memory usage at 1.20 GiB, and a bootdisk size of 32.00 GiB. The 'Notes' section is currently empty.

VMOKE (Uptime: 00:09:30)	
Status	running
HA State	none
Node	fauzi
CPU usage	1.50% of 4 CPU(s)
Memory usage	9.85% (403.45 MiB of 4.00 GiB)
Host memory usage	1.20 GiB
Bootdisk size	32.00 GiB
IPs	No Guest Agent configured

Pemantauan performa dilakukan melalui console mesin virtual pada Proxmox VE dengan menggunakan utilitas htop. Parameter yang diamati meliputi penggunaan CPU, memori, serta proses yang berjalan pada mesin virtual.

Berdasarkan hasil pemantauan, penggunaan CPU berada pada kondisi rendah hingga sedang dengan nilai load average sebesar 0.29, 0.59, dan 0.30. Kondisi ini menunjukkan bahwa proses pembuatan snapshot dan pengujian rollback tidak memberikan beban yang signifikan terhadap kinerja mesin virtual.

Penggunaan memori tercatat sekitar 172 MB dari total 3.82 GB, yang menunjukkan bahwa pemanfaatan RAM masih rendah dan sistem memiliki kapasitas yang cukup untuk menjalankan layanan tambahan. Tidak ditemukan proses yang menggunakan sumber daya secara berlebihan setelah implementasi backup snapshot.

Dengan demikian, dapat disimpulkan bahwa implementasi backup snapshot dan pengujian rollback dapat dilakukan tanpa menurunkan performa mesin virtual secara signifikan.

BAB VI

KESIMPULAN

Berdasarkan hasil implementasi yang telah dilakukan, fitur backup snapshot pada Proxmox VE berhasil diterapkan dengan baik. Snapshot mampu menyimpan kondisi mesin virtual secara point-in-time dan dapat digunakan untuk melakukan proses rollback ketika terjadi perubahan data, yang dibuktikan melalui pengujian pembuatan folder uji rollback dan file test.txt.

Selain itu, hasil pemantauan performa menunjukkan bahwa proses pembuatan snapshot dan pengujian rollback tidak memberikan dampak signifikan terhadap kinerja mesin virtual. Penggunaan CPU dan memori tetap berada pada kondisi normal, sehingga sistem tetap berjalan stabil setelah implementasi backup snapshot dilakukan.

Dengan demikian, dapat disimpulkan bahwa implementasi backup snapshot pada Proxmox VE efektif digunakan sebagai mekanisme pengamanan data sekaligus tidak mengganggu performa mesin virtual selama proses implementasi sistem.

DAFTAR PUSTAKA

Bobi Agustian , Dodi Susanto (2018)

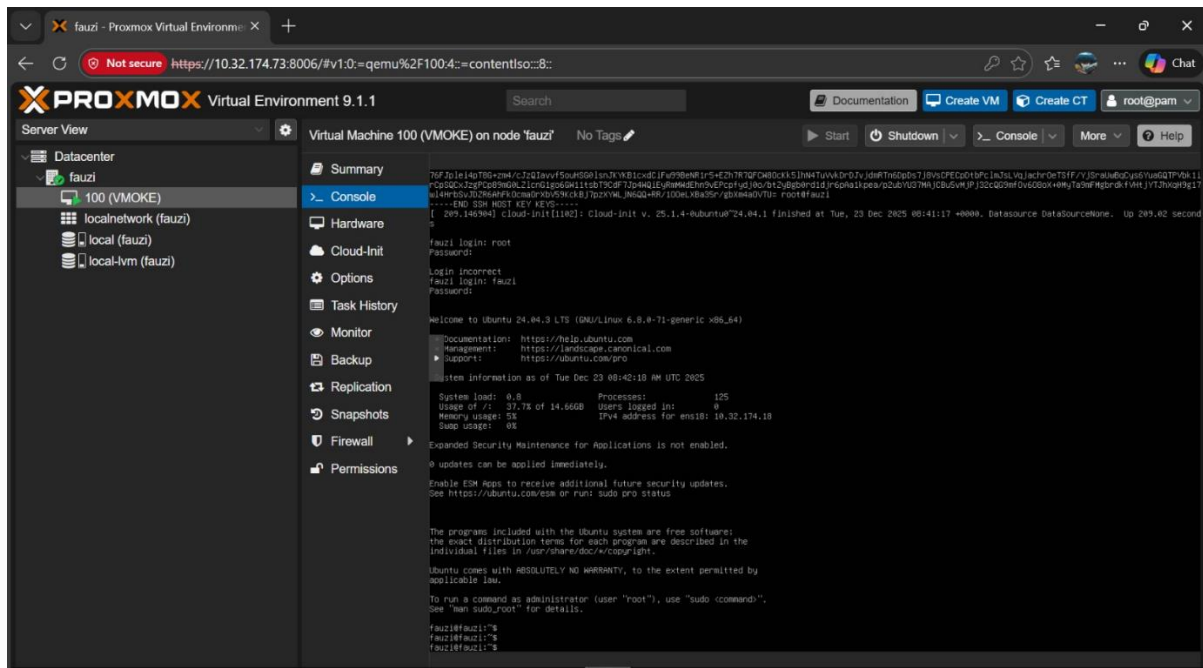
“VIRTUALISASI SERVER DENGAN PROXMOX UNTUK MENGOPTIMALKAN PENGGUNAAN SERVER” Jurnal Teknik Informatika ISSN 2549-4805 Volume 3, Nomor 2, Jul - Okt 2018

Rahmadi Rakhman, Nur Ayu Khalifah , Wahyudi Antoni (2024)

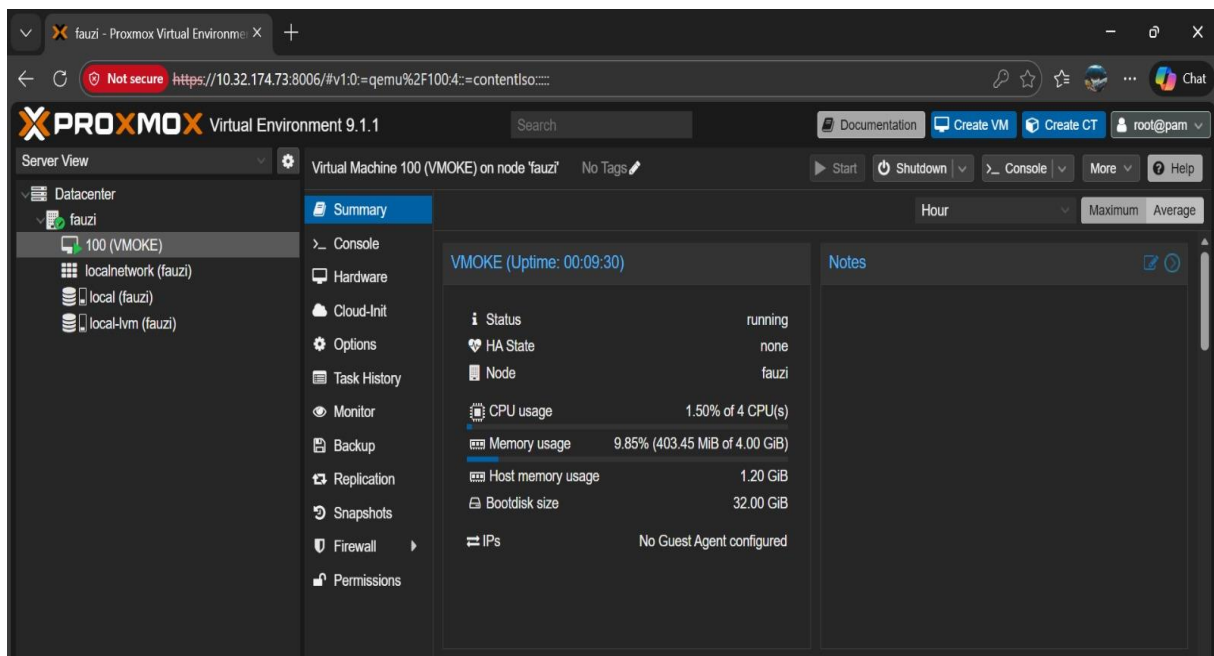
“Teknik Kontainerisasi dan Virtualisasi Sebagai Solusi Untuk Meningkatkan Portabilitas dan Skalabilitas Sistem Operasi Ubuntu” Jurnal Ilmiah Multidisiplin Volume 2, Nomor 8, Agustus 2024,

LAMPIRAN

A. Screenshot Setelah Konfigurasi Ubuntu Live Server



B. Summary Peforma dari Website Proxmox



C. Bukti Kerja

Minggu Pertama



Minggu Kedua



Minggu Ketiga



BONUS

