

Otomasi Manajemen dan Pengawasan Linux Container (LCX) Pada Proxmox VE Menggunakan Ansible

Linux Container Management and Monitoring Automation (LCX) on Proxmox VE Using Ansible

Muh. Akromi Arya Pratama^{1*}, I Putu Hariyadi²

^{1,2}Jurusan Ilmu Komputer, Universitas Bumigora

akromiaryapratama8@gmail.com^{1*}, putu.hariyadi@stmikbumigora.ac.id²

Abstrak – Penerapan sistem otomasi menggunakan ansible dapat mengefisienkan proses tersebut yang dilakukan secara manual. Ansible merupakan mesin otomatisasi Teknologi Informasi (TI) sederhana yang dapat mengotomatisasi cloud provisioning, manajemen konfigurasi, penerapan aplikasi, intra-service orchestration dan kebutuhan TI lainnya. Fitur yang diotomasi adalah pembuatan, menjalankan, menghentikan dan menghapus CT. dan juga pembuatan, mereset dan menghapus user dan permission akun dengan skenario ujicoba pada setiap perangkat yang terlibat berdasarkan rancangan ujicoba dan analisa terhadap hasil ujicoba yang telah dilakukan. Metode penelitian yang digunakan adalah prototype. Tahapan-tahapan pada metode prototype yaitu pengumpulan kebutuhan, membuat prototype, evaluasi prototype, pengkodean dan uji coba sistem. Hasil dari penelitian ini berupa sistem yang dapat mengotomasi pembuatan, menjalankan, menghentikan dan menghapus CT. dan juga mengotomasi pembuatan, mereset dan menghapus user dan permission. Manajemen CT, user dan permission dengan otomasi menggunakan tool ansible juga membuat proses manajemen CT, user dan permission menjadi lebih efisien dibandingkan dilakukan secara manual. Ansible dapat digunakan untuk mengotomasi manajemen CT baik dalam pembuatan, menjalankan, menghentikan dan menghapus CT pada PVE dan juga Ansible dapat digunakan untuk mengotomasi manajemen user & permission baik dalam pembuatan, mereset dan menghapus user & permission pada PVE. dalam mata pelajaran administrasi system jaringan di SMKN 6 Mataram.

Kata Kunci: Otomasi, CT, Ansible, Proxmox Virtual Environment.

Abstract – The implementation of the automation system uses processes that can be carried out efficiently which are carried out manually. Enabling is a simple Information Technology (IT) automation engine that can automate cloud provisioning, configuration management, application implementation, intra-service orchestration and other IT needs. The automated features are creating, running, stopping and installing CT. and also creates, resets and installs user and account permissions with a trial scenario on each device involved based on the trial design and analyzes the results of the trials that have been carried out. The research method used was a prototype. The stages in the prototype method are planning requirements, making prototypes, evaluating prototypes, coding and testing systems. The results of this study consist of a system that can automate the manufacturing, running, stopping and installing CT. and also automates creation, resetting and installing user permissions and. CT management, users and permits with automation using tools might also make the CT, user and permission management process more efficient than done manually. Can be used to automate CT management both in manufacturing, running, stopping and installing CT in PVE and also in the network administration system subjects at SMK 6 Mataram.

Keywords: Automation, CT, Ansible, Proxmox Virtual Environment.

Jurnal Bumigora Information Technology (BITE)

Vol.3, No.1, Juni 2021, pp. 82-95

ISSN: 2685-4066

DOI: 10.30812/bite.v3i1.807

1. Pendahuluan

Sekolah Menengah Kejuruan Negeri (SMKN) 6 Mataram merupakan salah satu sekolah yang beralamat di Jl. Lalu Mesir, Babakan, Kota Mataram. SMKN 6 Mataram memiliki 7 (tujuh) jurusan yaitu Teknik Jaringan dan Komputer (TKJ), Teknik Perbaikan Bodi Otomotif, Teknik Pengelasan, Teknik kendaraan Ringan, Teknik pendingin dan tata udara, Bisnis konstruksi dan property, dan Desain grafis. Pihak sekolah menyediakan laboratorium komputer dan perangkat jaringan untuk menunjang kegiatan praktikum pada program studi TKJ. Selain itu untuk mendukung praktikum dari salah satu mata pelajaran yaitu administrasi system jaringan, guru dari program studi TKJ juga telah memanfaatkan media virtualisasi berbasis *Proxmox Virtual Environment*[1][2].

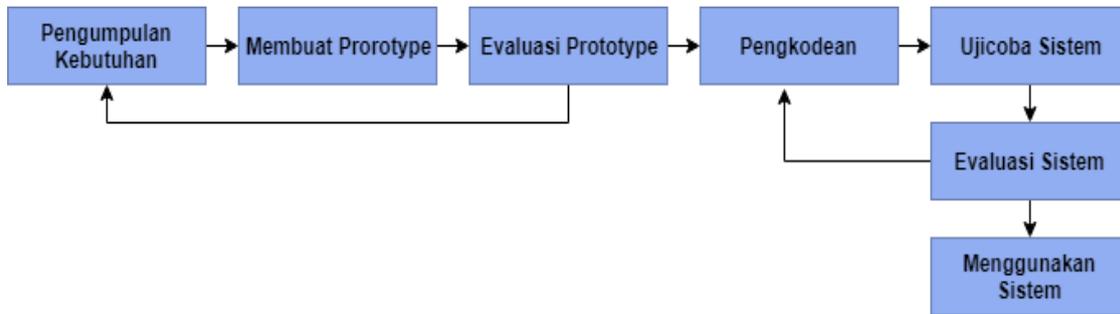
Secara umum, sebelum dapat memanfaatkan media pembelajaran *virtualisasi* berbasis *PVE* tersebut maka diperlukan pengaturan yang harus di persiapkan oleh pihak guru agar siswa dapat menggunakan *PVE* sebagai media pembelajaran[3]–[5]. Untuk guru terdapat beberapa hal yang perlu dipersiapkan terlebih dahulu, seperti pembuatan *container (CT)*[6], [7], *user*, dan pengaturan *permission* yang di lakukan di setiap awal semester, dibutuhkan *monitoring CT* yang sedang aktif dan tidak aktif saat pertemuan setiap minggunya, dan penghapusan *CT*, *user* dan *permission* yang dilakukan di setiap akhir semester. Untuk saat ini Teknik yang digunakan untuk menyiapkan media pembelajaran berbasis *Proxmox VE* masih dilakukan secara manual melalui *Graphical User Interface (GUI)* dari *Proxmox VE*.

Dari permasalahan yang ada dikarenakan masih dilakukan secara manual dan membutuhkan waktu yang cukup lama sehingga untuk proses pembelajaran dirasa kurang efisien, dari pihak sekolah khususnya guru yang mengampu mata pelajaran tersebut memiliki harapan adanya sistem yang dapat mengotomasi pembuatan *CT*, *user* dan pengaturan *permission* sehingga setiap memasuki tahun ajaran baru di awal semester guru dapat membuat *CT*, *user* dan pengaturan *permission* sejumlah siswa yang mengampu mata pelajaran tersebut, selama satu semester di pertemuan setiap minggunya mata pelajaran berlangsung saat proses praktikum dapat dimonitoring *CT* yang aktif, dan di akhir semester dapat menghapus *CT*, *user* dan *permission* saat siswa yang mengampu mata pelajaran tersebut lulus dari mata pelajaran tersebut. Sehingga semuanya dapat diakomodir oleh guru yang mengampu mata pelajaran tersebut baik pembuatan *CT*, *User* dan pengaturan *permission*, *monitoring CT* dan menghapus *CT*, *user* dan *permission* tanpa dilakukan dengan cara manual lagi.

Otomasi manajemen dan pengawasan *CT* pada *PVE* menggunakan *Ansible* dapat membantu mengatasi permasalahan yang dihadapi oleh guru terkait manajemen dan pengawasan *CT* pada *PVE* tersebut. Menurut Kamus Besar Bahasa Indonesia (2019), otomatisasi adalah penggantian tenaga manusia dengan tenaga mesin yang secara otomatis melakukan dan mengatur pekerjaan sehingga tidak memerlukan lagi pengawasan manusia (dalam industri dan sebagainya). *Ansible* merupakan mesin otomatisasi Teknologi Informasi (TI) sederhana yang dapat mengotomatisasi *cloud provisioning*, manajemen konfigurasi, penerapan aplikasi, *intra-service orchestration* dan kebutuhan TI lainnya.

Dengan adanya penelitian ini diharapkan dapat memberikan manfaat berupa pengetahuan dan wawasan dalam memanajemen dan pengawasan sistem *virtualisasi* otomasi pada *PVE* sehingga proses pembelajaran lebih efektif dan efisien.

2. Metode Penelitian



Gambar 1 Metodologi *Prototype*

Metodologi penelitian yang digunakan pada penelitian ini yaitu *prototype*. Dari tujuh tahapan yang ada, Terdapat lima dari tujuh tahapan yang digunakan yaitu Tahap Pengumpulan Kebutuhan, Membuat *Prototype*, Evaluasi *Prototype*, Pengkodean, Ujicoba Sistem.

2.1. Pengumpulan Kebutuhan

2.1.1 Pengumpulan Data

Terdapat 3 metode pengumpulan data yang digunakan yaitu observasi, wawancara dan dokumentasi. Adapun penjelasan dari masing-masing metode pengumpulan data yang digunakan adalah sebagai berikut:

A. Observasi

Metode observasi digunakan untuk mengumpulkan data dengan cara melakukan pengamatan secara langsung pada SMKN 6 Mataram sehingga dapat diperoleh informasi terkait sistem yang telah berjalan seperti perangkat jaringan, infrastruktur jaringan dan sistem yang digunakan. Dari hasil observasi yang telah dilakukan pada SMKN 6 Mataram maka diperoleh informasi sebagai berikut:

- a) SMKN 6 Mataram memiliki ruang laboratorium komputer yang di peruntukan untuk mata pelajaran administrasi system jaringan yang didalamnya memuat perangkat infrastruktur jaringan, *server* dan komputer *client*.
- b) *Server* telah menerapkan *virtualisasi* menggunakan *PVE* untuk mendukung proses pembelajaran administrasi system jaringan.
- c) Terdapat server 24 komputer *client* pada laboratorium TKJ SMKN 6 Mataram yang terhubung ke *server*.

B. Wawancara

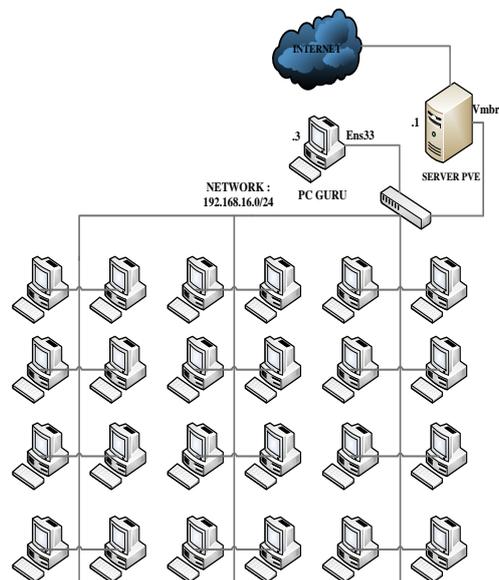
Metode pengumpulan data menggunakan wawancara dilakukan untuk memperoleh informasi lebih detail terkait sistem yang telah berjalan, kendala-kendala yang dihadapi, harapan dari SMKN 6 Mataram sehingga dapat ditentukan solusi penyelesaiannya. Wawancara dilakukan dengan Guru pengampu mata pelajaran administrasi system jaringan yaitu dengan Bapak I Gusti Bagus Wirayuda, A.Md. Daftar pertanyaan dan jawaban dari wawancara, seperti terlihat pada tabel 1.

Tabel 1 Wawancara Dengan Guru Pengampu Mata Pelajaran Administrasi System **Jaringan**

	Pertanyaan	Jawaban
1	Berapa jurusan yang ada di SMKN 6 Mataram saat ini?	Di SMKN 6 Mataram ada 7 jurusan yaitu Teknik Jaringan dan Komputer, Teknik Perbaikan Bodi Otomotif, Teknik Pengelasan, Teknik kendaraan Ringan, Teknik Pendingin dan Tata Udara, Bisnis Konstruksi dan Properti, dan Desain Grafis.
2	Terkait dengan prodi TKJ kendala-kendala apa saja yang dihadapi saat proses belajar mengajar?	Dalam proses praktikum belum ada manajemen terpusat dalam <i>virtualisasi</i> dan kekurangan perangkat pembelajaran dalam lab.
3	Berdasarkan kurikulum yang digunakan mata pelajaran apa yang membutuhkan praktikum?	Salah satu mata pelajaran yang sangat membutuhkan praktikum adalah mata pelajaran administrasi sistem jaringan.
4	Apakah ada RPP dan modul untuk mendukung proses pembelajaran praktikum administrasi jaringan?	Ada, RPP yang terselenggara pada semester ganjil dan genap.
5	Berapa laboratorium dan berapa computer yang diperuntukan untuk mata pelajaran praktikum admistrasi system jaringan?	Ada 1 LAB yang diperuntukan untuk mata pelajaran administrasi system jaringan dengan 1 komputer sebagai <i>server</i> dan 24 komputer sebagai <i>client</i> .
6	Ada berapa jumlah siswa yang mengikuti praktikum administrasi system jaringan?	Ada 24 siswa yang mengikuti mata pelajaran administrasi system jaringan.
7	Apakah proses belajar mengajar praktikum administrasi system jaringan telah menggunakan virtualisasi seperti <i>VMware</i> , <i>Virtualbox</i> atau yang lain?	Iya, selama ini proses belajar mengajar praktikum administrasi system jaringan menggunakan virtualisasi seperti <i>VMware Workstation</i> dan <i>Proxmox VE</i> .

8	Pada proses belajar mengajar praktikum administrasi sistem jaringan menggunakan virtualisasi <i>Proxmox</i> , apakah ada kendala atau permasalahan yang dialami?	Kendala pada pembuatan CT, user, pengaturan permission dan monitoring user yang masih dilakukan secara manual di setiap semesternya.
9	Apakah ada kendala terkait sistem yang tepat untuk mempercepat pembuatan CT, <i>User dan pengaturan permission</i> pada server <i>Proxmox</i> ?	Iya masih terkendala menemukan system yang tepat untuk mengatasi kendala tersebut.
10	Harapan apa untuk kedepan terkait permasalahan ini?	Diharapkan adanya system yang tepat yang dapat mengotomasi pembuatan <i>CT, user, pengaturan permission</i> dan adanya system <i>monitoring</i> di setiap semesternya.

C. Dokumentasi



Gambar 2 Topologi LAB TKJ Pada SMKN 6 Mataram

Metode Metode pengumpulan data melalui dokumentasi dilakukan untuk memperoleh dokumen terkait rancangan topologi jaringan yang telah diterapkan pada SMKN 6 Mataram. Topologi jaringan yang digunakan oleh SMKN 6 Mataram seperti terlihat pada gambar 2.

Pada laboratorium komputer TKJ, terdapat 1 buah server PVE, dengan 24 *client* untuk siswa dan 1 *PC* untuk guru, yang terhubung langsung pada *switch*.

2.2. Membangun Prototype

Tahap ini memuat tentang rancangan jaringan lama, rancangan jaringan alternatif, rancangan uji coba, rancangan sistem otomasi, rancangan pengelamatan IP, serta kebutuhan perangkat keras dan perangkat lunak berdasarkan dari hasil analisa data pada tahap sebelumnya.

2.2.1. Rancangan Jaringan Lama

Rancangan jaringan lama berisikan, topology jaringan yang digunakan saat ini di SMKN 6 Mataram, rancangan jaringan tersebut merupakan gambaran sebelum diterapkannya system otomasi. Berikut ini merupakan rancangan jaringan lama yang digunakan.

Rancangan jaringan lama berisikan *server PVE* yang terhubung langsung ke internet, dengan 1 komputer guru sebagai *administrator* dan terdapat 24 *client* yang terhubung melalui *switch*.

2.2.2. Rancangan Sistem Manual PVE

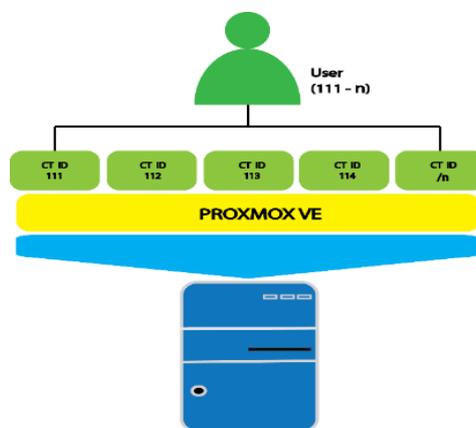
Rancangan sistem manual PVE berisikan rancangan yang akan digunakan dalam pembuatan *CT* pada *server PVE*. Seperti terlihat pada gambar 6.

Dalam pembuatan akun pada *CT*, digunakan data siswa berupa NIS (Nomor Induk Siswa) sebagai pengenalan akun pada *CT* untuk masing-masing siswa. Data tersebut dapat dilihat seperti pada tabel berikut.

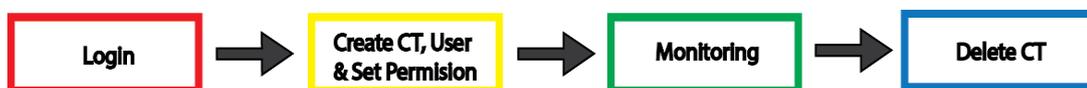
Tabel 2 Data Pembuatan Akun Otentikasi

User Menggunakan NIS (Nomor Induk Siswa)	CT ID	Permission
17021816	111	PVEVM Admin
17021818	112	
17021819	113	
17021820	114	
n	n	

Pembuatan *CT*, *user* dan pengaturan *permission*, *monitoring* dan penghapusan *CT* pada *server PVE* sebelum diterapkannya otomasi memiliki tahapan-tahapan seperti pada gambar 7.



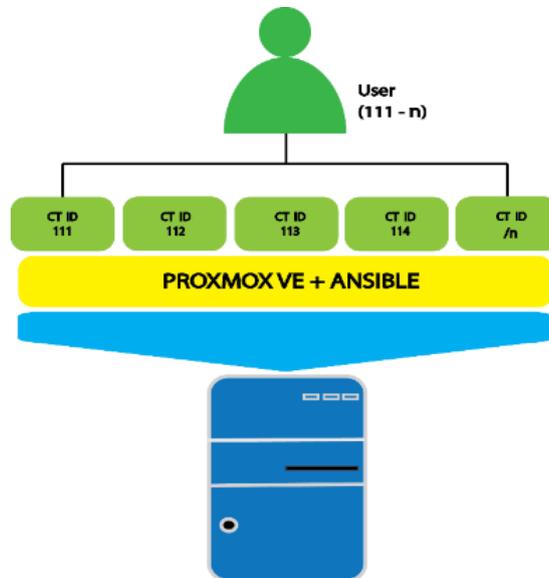
Gambar 3 Rancangan Sistem Manual PVE



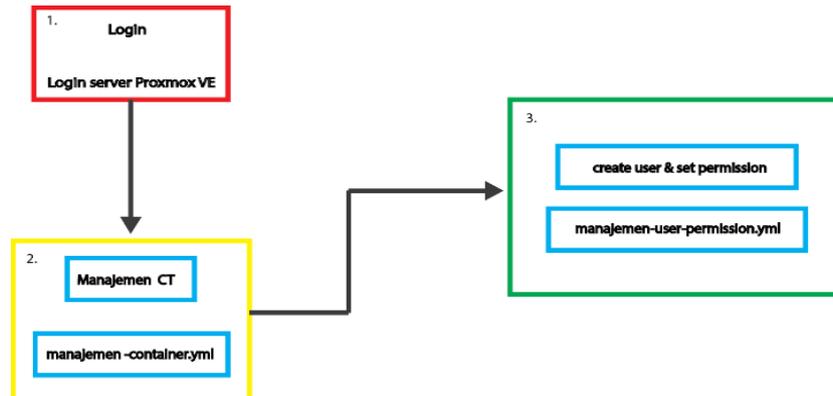
Gambar 4 Tahapan Pembuatan Sistem Manual Pada PVE

2.2.3. Rancangan Sistem Otomasi Proxmox Menggunakan Ansible

Rancangan Rancangan sistem otomasi merupakan rancangan yang akan digunakan untuk melakukan otomasi untuk memanajemen *CT*, *user* dan pengaturan *permission* pada *server PVE* berdasarkan data siswa. Dalam hal ini *server PVE* sudah terintegrasi dengan *ansible*, sehingga memanajemen *CT*, *user* dan pengaturan *permission* dapat dibuat secara otomasi. Seperti pada gambar 6.



Gambar 5 Rancangan Sistem Otomasi PVE



Gambar 6 Tahapan Manajemen CT, User dan Permission Secara Otomasi Pada PVE

Dalam memanajemen *CT*, *User* dan pengaturan *permission* secara otomasi, terdapat tahapan-tahapan yang dilakukan agar proses otomasi dapat berjalan. Tahapan tersebut dapat dilihat pada gambar 7 diatas.

Tahapan pertama yang dilakukan adalah dengan *login ke server Ansible*, kemudian untuk melakukan proses memanajemen *CT* dengan mengakses file *YAML* dengan nama *manajemen-container.yml*. Setelah itu mengisikan script untuk memanajemen *CT* pada *PVE* untuk mengeksekusi file *YAML* dengan mengetikan perintah *ansible-playbook manajemen-container.yml --skip-tags "start, stop, delete"* karna ada beberapa *tags* yang kita jalankan saat memanajemen *container*. Selanjutnya memanajemen *user* dan *permission* dengan mengakses file *YAML* dengan nama *manajemen-user-permission.yml*. Setelah itu mengisikan script untuk memanajemen *user* dan *permission* pada *PVE* untuk mengeksekusi file *YAML* dengan

mengetikan perintah *ansible-playbook manajemen-user-permission.yml --skip-tags "remove, reset"*

2.2.4. Kebutuhan Perangkat Keras Dan Lunak

Adapun kebutuhan perangkat keras dan perangkat lunak dalam penulisan skripsi ini adalah sebagai berikut:

A. Kebutuhan Perangkat Keras (Hardware)

1. Dengan menggunakan 1 (satu) unit computer yang didalamnya diinstallkan *VMware workstation* dan dibuatkan 2 (dua) *virtual machin* sebagai server PVE dan *server ansible*. Dan juga 1 (satu) unit computer itu juga akan difungsikan sebagai *PC Client*.
 - a. 1 *server PVE* dengan spesifikasi sebagai berikut:
 - Procesor : 1 Core
 - Hardisk : 40 GB
 - Memory : 4 GB
 - b. 1 *server asnible* dengan spesifikasi sebagai berikut:
 - Procesor : 1 Core
 - Hardisk : 15 GB
 - Memory : 2 GB
 - Sistem Operasi : CentOS 7
 - c. Komputer Client dengan spesifikasi sebagai berikut:
 - Procesor : Intel Core i5
 - Hardisk : 1 TB
 - Memory : 8 GB
 - Sistem Operasi : Windows 10

B. Kebutuhan Perangkat Lunak (Software)

1. *Ansible* merupakan mesin otomatisasi Teknologi Informasi (TI) sederhana yang dapat mengotomatisasi *cloud provisioning*, manajemen konfigurasi, penerapan aplikasi, *intra-service orchestration* dan kebutuhan TI lainnya. *Ansible* diinstall pada *PC Guru* yang digunakan sebagai *server ansible* untuk melakukan otomasi.
2. *Pip* adalah singkatan dari *Pip Installs Python* atau *PIP Installs Packages*. *pip* adalah sebuah *app store* (atau biasa disebut sebagai package manager), kita bisa mencari, menginstall, *manage modules* atau *package* pada instalasi *python*. *Pip* terinstal pada *server PVE*.
3. *Template CentOS 7* sebagai *OS* pada computer *server* dan digunakan untuk instalasi sistem operasi pada *CT*.
4. *Web browser* adalah suatu program atau *software* yang digunakan untuk menjelajahi internet atau untuk mencari informasi dari suatu *web* yang tersimpan didalam komputer. *Web browser* digunakan untuk mengakses halaman *dashboard web PVE* melalui *client*.

C. Evaluasi Prototype

Pada tahap evaluasi, berisikan pernyataan penerimaan dan persetujuan sistem otomasi yang telah di ajukan kepada SMKN 6 Mataram.

D. Pengkodean Sistem

1. Instalasi Dan Konfigurasi

Instalasi dan konfigurasi

Pada tahap ini memuat tentang instalasi dan konfigurasi yang dilakukan pada masing-masing perangkat yang terlibat berdasarkan rancangan uji coba.

- a. Instalasi dan konfigurasi *server PVE*
 - Konfigurasi pengalamatan IP pada server

- Install *python-pip*
 - Install *proxmoxer*
- b. Instalasi dan konfigurasi *server Ansible*
- Install *epel-release*
 - Install *ansible*
- c. Konfigurasi pada *client*
- Konfigurasi pada *client* meliputi konfigurasi pengalamatan TCP/IP agar *client* dapat mengakses dashboard web PVE.
2. Pembuatan Kode Program
- Pada tahap ini dilakukan pembuatan kode-kode program yang terkait dengan sistem otomasi yaitu:
- a. pembuatan file manajemen CT dengan nama “*manajemen-container.yml*”
 - b. pembuatan file manajemen User dan Permission “*manajemen-user-permission.yml*”

E. Menguji Sistem

Pada tahap ini memuat tentang langkah-langkah untuk uji coba hasil konfigurasi menggunakan beberapa skenario uji coba. Dalam melakukan skenario uji coba manajemen *CT* seperti pembuatan, menjalankan, menghentikan dan menghapus *CT* dan manajemen *user* dan *permission* seperti pembuatan, mereset dan menghapus *user* dan *permission*. baik secara manual maupun otomasi, digunakan pembuatan menggunakan 1 data siswa sebanyak 5 kali dan menggunakan 24 data siswa sebanyak 5 kali, serta melakukan perhitungan waktu yang dibutuhkan untuk manajemen *CT* dan manajemen *user* dan *permission*.

Tabel 3 Skenario Ujicoba

Jumlah Ujicoba	Jenis pembuatan	Deskripsi
5 kali, dengan menggunakan 1 data siswa.	Manual (melalui <i>dashboard web PVE</i>)	Menghitung waktu minimum, maksimum, dan rata-rata pembuatan, menjalankan, menghentikan, menghapus <i>CT</i> dan pembuatan, menghapus dan mereset semua <i>password user</i> dan <i>permission</i> .
5 kali, dengan menggunakan 24 data siswa	Otomasi (pembuatan menggunakan <i>ansible</i>)	Menghitung waktu minimum, maksimum, dan rata-rata pembuatan, menjalankan, menghentikan, menghapus <i>CT</i> dan pembuatan, menghapus dan mereset semua <i>password user</i> dan <i>permission</i> .
5 kali, menggunakan 1 data siswa.	Manual (melalui <i>dashboard web PVE</i>)	Menghitung waktu minimum, maksimum, dan rata-rata pembuatan, menjalankan, menghentikan, menghapus <i>CT</i> dan pembuatan, menghapus dan mereset semua <i>password user</i> dan <i>permission</i> .
5 kali, menggunakan 24 data siswa.	Otomasi (pembuatan menggunakan <i>ansible</i>)	Menghitung waktu minimum, maksimum, dan rata-rata pembuatan, menjalankan, menghentikan, menghapus <i>CT</i> dan pembuatan, menghapus dan mereset semua

		<i>password user dan permission.</i>
--	--	--------------------------------------

3. Hasil dan Pembahasan

3.1. Hasil Instalasi dan Konfigurasi

Pada server dilakukan instalasi sistem operasi *Proxmox Virtual Environment (PVE)* dengan hasil instalasi seperti terlihat pada gambar 10.

```

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

https://192.168.16.1:8006/

-----

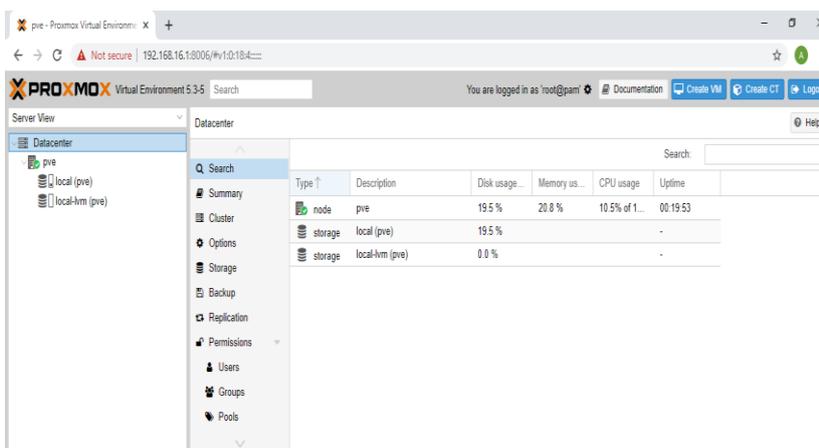
pve login: root
Password:
Last login: Mon Jan 6 21:05:22 WITA 2020 on tty1
Linux pve 4.15.18-9-pve #1 SMP PVE 4.15.18-30 (Thu, 15 Nov 2018 13:32:46 +0100) x86_64

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.
root@pve:~#
    
```

Gambar 10 Hasil Instalasi Proxmox VE

Gambar 11 Tampilan Login PVE Web



Gambar 12 Tampilan Dashboard Web Server Proxmox VE

Terlihat kotak dialog otentikasi Proxmox VE *Login*. Login menggunakan *browser* ke *server* Proxmox VE dengan alamat IP `https://192.168.16.1:8006` dan lengkapi isian dari “User name” dan “Password”. Pada isian “User name” masukan “root”. Sedangkan pada isian “Password” masukan sandi login dari user “root” yaitu “12345678”. Selain itu pastikan pilihan “Realm” adalah Linux PAM standard authentication serta untuk “Language” adalah “English”, seperti gambar 11.

Terlihat pada PVE yang telah di install dan terdapat menu-menu lainnya yang membantu dalam memmanagement ketika menggunakan *server* PVE sebagai management *virtualization*, seperti terlihat pada gambar 12.

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

ansible login: root
Password:
Last login: Mon Jan 6 14:43:58 from 192.168.16.2
root@ansible ~]# ansible --version
ansible 2.9.1
  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)]
root@ansible ~]# _
```

Gambar 13 Tampilan Hasil Instalasi *Ansible*

```
pve login: root
Password:
Last login: Mon Jan 6 21:36:30 WITA 2020 on tty1
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.
root@pve:~# pip --version
pip 9.0.1 from /usr/lib/python2.7/dist-packages (python 2.7)
root@pve:~# _
```

Gambar 14 Tampilan Hasil Instalasi *Package PIP*

```
[servers]
pve ansible_host=192.168.16.1
```

Gambar 15 Tampilan Hasil Konfigurasi *File Hosts*

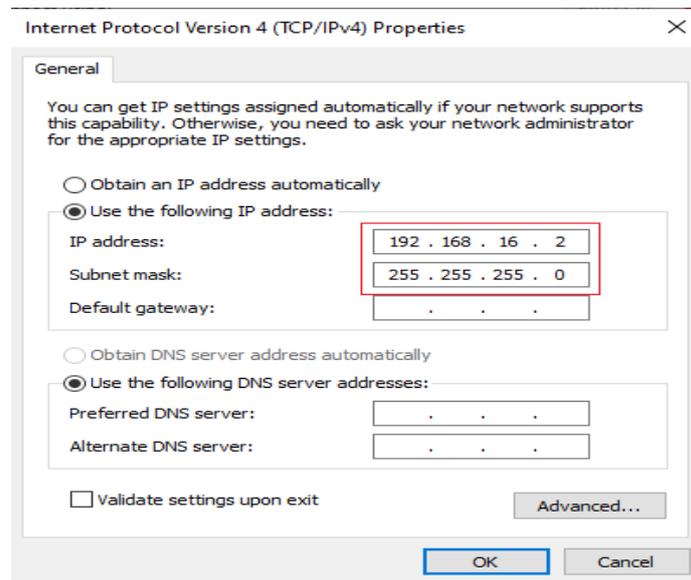
Selanjutnya Pada *Server Linux CentOs 7* dilakukan instalasi *tools ansible* yang digunakan untuk melakukan otomasi pada *proxmox* dan juga melakukan konfigurasi file *host*. Hasil instalasi dan konfigurasi *ansible* terdiri dari instalasi *epel-release* dan instalasi *package ansible*. Hasil akhir dari instalasi *tools ansible* pada server PVE, terlihat seperti pada gambar 13.

Terdapat *package pip* yang dibutuhkan *ansible* agar sistem otomasi pada server *proxmox* dapat berjalan, hasil dari instalasi *package* seperti terlihat pada gambar 14

Setelah instalasi *package* pada server *proxmox* dan *ansible* selesai, selanjutnya melakukan konfigurasi file *hosts* yang terdapat pada `/etc/ansible/hosts` dengan menambahkan ip dari server *proxmox* dengan alamat IP `192.168.16.1` agar *ansible* dapat berkomunikasi dengan *proxmox*, hasil dari konfigurasi file *hosts*, seperti terlihat pada gambar 15 berikut:

3.2. Hasil Konfigurasi Client Windows 10

Konfigurasi yang dilakukan pada client *windows 10* adalah pengaturan alamat IP agar dapat terhubung dengan server *proxmox* Terlihat alamat IP yang digunakan adalah `192.168.16.2` dengan subnetmask `255.255.255.0`. Hasil konfigurasi IP seperti terlihat pada gambar 16.



Gambar 16 Tampilan Hasil Konfigurasi Client

3.3. Analisa Hasil Ujicoba

Berdasarkan analisa hasil ujicoba yang telah dilakukan maka diperoleh analisa hasil ujicoba sebagai berikut:

1. Pembuatan, menjalankan, menghentikan dan menghapus *CT*. secara umum antara manual dengan otomasi berbeda. Pembuatan, menjalankan, menghentikan dan menghapus *CT* secara manual hanya bisa dilakukan 1 (satu) kali dalam satu waktu. Sedangkan pembuatan, menjalankan, menghentikan dan menghapus *CT* secara otomasi bisa membuat lebih dari 1 (satu) *CT* dalam satu waktu secara bersamaan atau dalam satu kali eksekusi perintah menggunakan *ansible*.
2. Pembuatan, mereset, dan menghapus *user & Permission*. secara umum antara manual dengan otomasi berbeda. Pembuatan, mereset, dan menghapus *user & Permission* secara manual hanya bisa dilakukan 1 (satu) kali dalam satu waktu. Sedangkan pembuatan, mereset, dan menghapus *User & Permission* secara otomasi bisa membuat lebih dari 1 (satu) *user & Permission* dalam satu waktu secara bersamaan atau dalam satu kali eksekusi perintah menggunakan *ansible*.
3. Analisa waktu yang dibutuhkan untuk Pembuatan, menjalankan, menghentikan dan menghapus *CT*. Terdapat dua analisa perbandingan hasil ujicoba waktu Pembuatan, menjalankan, menghentikan dan menghapus *CT*. Pertama yaitu analisa hasil ujicoba antara waktu Pembuatan, menjalankan, menghentikan dan menghapus 1 (satu) *CT* manual dan otomasi. Kedua yaitu analisa hasil Pembuatan, menjalankan, menghentikan dan menghapus 24 (dua puluh empat) *CT* sekaligus baik secara manual maupun secara otomasi menggunakan *ansible*. Hasil analisa yaitu perhitungan waktu sebanyak 5 (lima) kali percobaan secara manual dan otomasi, dapat terlihat rata-rata waktu pembuatan, menjalankan, menghentikan dan menghapus 1 (satu) *CT*. Selisih dari waktu pembuatan adalah 21 detik, selisih dari waktu menjalankan adalah 2 detik, Selisih dari waktu menghentikan adalah 3 detik dan selisih dari waktu penghapusan adalah 1 detik lebih cepat dilakukan secara otomasi. Perbedaan waktu juga dipengaruhi oleh administrator ketika memasukkan data yang akan diotomasi (human error). Dari hasil analisa perhitungan waktu sebanyak 5 (lima) kali percobaan secara manual dan otomasi, dapat terlihat rata-rata pembuatan, menjalankan, menghentikan dan menghapus 24 (dua puluh empat) *CT*. Selisih dari waktu pembuatan adalah 15 menit 14 detik, selisih dari waktu menjalankan adalah 02 menit 06 detik, Selisih dari waktu menghentikan adalah 03 menit 18 detik dan selisih dari

waktu penghapusan adalah 02 menit 39 detik lebih cepat dilakukan secara otomatis. Perbedaan waktu juga dipengaruhi oleh administrator ketika memasukkan data yang akan diotomasi (*human error*).

4. Analisa waktu yang dibutuhkan untuk Pembuatan, mereset dan menghapus *user & permission*. Terdapat dua analisa perbandingan hasil ujicoba waktu Pembuatan, mereset dan menghapus *user & permission*. Pertama yaitu analisa hasil ujicoba antara waktu Pembuatan, mereset dan menghapus 1 (satu) *user & permission* manual dan otomatis. Kedua yaitu analisa hasil Pembuatan, mereset dan menghapus 24 (dua puluh empat) *user & permission* sekaligus baik secara manual maupun secara otomatis menggunakan *ansible*. Hasil analisa seperti berikut ini:
 - a. Dari hasil analisa perhitungan waktu sebanyak 5 (lima) kali percobaan secara manual dan otomatis, dapat terlihat rata-rata pembuatan, mereset, dan menghapus 1 (satu) *user & permission* lebih cepat dilakukan secara otomatis. Perbedaan waktu juga dipengaruhi oleh administrator ketika memasukkan data yang akan diotomasi (*human error*).
 - b. Dari hasil analisa perhitungan waktu sebanyak 5 (lima) kali percobaan secara manual dan otomatis, dapat terlihat rata-rata pembuatan, mereset, dan menghapus 24 (dua puluh empat) *user & permission*. Selisih dari waktu pembuatan adalah 5 menit 19 detik, selisih dari waktu mereset adalah 03 menit 18 detik dan selisih dari waktu penghapusan adalah 2 menit lebih cepat dilakukan secara otomatis. Perbedaan waktu juga dipengaruhi oleh administrator ketika memasukkan data yang akan diotomasi (*human error*).

4. Kesimpulan

Adapun kesimpulan yang dapat diambil berdasarkan hasil uji coba yang telah dilakukan adalah sebagai berikut:

1. *Ansible* dapat digunakan mengotomasi manajemen *CT* baik dalam pembuatan, menjalankan, menghentikan dan menghapus *CT* pada *PVE* dalam mata pelajaran administrasi system jaringan di SMKN 6 Mataram.
2. Sistem *Ansible* dapat digunakan untuk mengotomasi manajemen *user & permission* baik dalam pembuatan, mereset dan menghapus *user & permission* pada *PVE*.
3. Sistem otomatis yang dibuat dapat mempercepat proses pembuatan, menjalankan, menghentikan dan menghapus 1 (satu) *CT*. Dilakukan sebanyak 5 (lima) kali percobaan dengan rata-rata Selisih waktu pembuatan adalah 21 detik, selisih dari waktu menjalankan adalah 2 detik, selisih dari waktu menghentikan adalah 3 detik dan selisih dari waktu penghapusan adalah 1 detik lebih cepat dilakukan secara otomatis dari pada dilakukan secara manual.
4. Sistem otomatis yang dibuat dapat mempercepat proses pembuatan, menjalankan, menghentikan dan menghapus 24 (dua puluh empat) *CT*. Dilakukan sebanyak 5 (lima) kali percobaan dengan rata-rata Selisih waktu pembuatan adalah 15 menit 14 detik, selisih dari waktu menjalankan adalah 2 menit 6 detik, Selisih dari waktu menghentikan adalah 3 menit 18 detik dan selisih dari waktu penghapusan adalah 2 menit 39 detik lebih cepat dilakukan secara otomatis dari pada dilakukan secara manual.
5. Sistem otomatis yang dibuat dapat mempercepat proses pembuatan, mereset, dan menghapus 1 (satu) *user dan permission*. Dilakukan sebanyak 5 (lima) kali percobaan dengan rata-rata Selisih waktu pembuatan adalah 5 detik, selisih dari waktu mereset adalah 9 detik, dan selisih dari waktu penghapusan adalah 6 detik lebih cepat dilakukan secara otomatis dari pada dilakukan secara manual.
6. Sistem otomatis yang dibuat dapat mempercepat proses pembuatan, mereset, dan menghapus 24 (dua puluh empat) *user dan permission*. Dilakukan sebanyak 5 (lima) kali percobaan

dengan rata-rata Selisih waktu pembuatan adalah 5 menit 19 detik, selisih dari waktu mereset adalah 3 menit 18 detik, dan selisih dari waktu penghapusan adalah 2 menit lebih cepat dilakukan secara otomatis dari pada dilakukan secara manual.

Ucapan Terima Kasih

Terimakasih kepada tim jurnal BITe yang telah memberikan kesempatan pada kami untuk mempublikasikan penelitian yang kami lakukan.

Referensi

- [1] Arief Arfriandi, "Perancangan, implementasi, dan Analisis Kinerja Virtualisasi Menggunakan Proxmox Esx, Vmware dan Openstack," *Jurnal Teknologi*, vol. 5, no. 2, pp. 182–191, 2012.
- [2] Y. Ariyanto, B. Harijanto, V. A. H. Firdaus, and S. N. Arief, "Performance analysis of Proxmox VE firewall for network security in cloud computing server implementation," *IOP Conference Series: Materials Science and Engineering*, vol. 732, no. 1, pp. 1–6, 2020, doi: 10.1088/1757-899X/732/1/012081.
- [3] I. P. Hariyadi and A. Juliansyah, "Analisa Penerapan Private Cloud Computing Berbasis Proxmox Virtual Environment Sebagai Media Pembelajaran Praktikum Manajemen Jaringan," *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 18, no. 1, pp. 1–12, 2018, doi: 10.30812/matrik.v18i1.329.
- [4] S. R. Siregar, P. Studi, and T. Informatika, "Efisiensi Fisik Komputer Server dengan Menerapkan Proxmox Virtual Environment," *Journal of Computer System and Informatics (JoSYC)*, vol. 1, no. 2, pp. 83–87, 2020.
- [5] E. A. Didik Sudayana, "Virtualisasi Server Dengan Proxmox Untuk Pengoptimalisasian Penggunaan Resource Server Pada UPT Teknologi dan Komunikasi Pendidikan," *SATIN - Sains dan Teknologi Informasi*, vol. 3, no. 2, p. 8, 2014.
- [6] W. Felter, A. Ferreira, R. Rajamony, and J. Rubio, "IBM Research Report An Updated Performance Comparison of VirtualMachines and Linux Containers," *Computer Science*, vol. 25297, pp. 25482–1407, 2012, [Online]. Available: <http://domino.watson.ibm.com/library/CyberDig.nsf/home>.
- [7] Z. Wan, D. Lo, X. Xia, L. Cai, and S. Li, "Mining Sandboxes for Linux Containers," *Proceedings - 10th IEEE International Conference on Software Testing, Verification and Validation, ICST 2017*, pp. 92–102, 2017, doi: 10.1109/ICST.2017.16.