

# Implementasi *Ansible Playbook* untuk Mengotomatisasi Manajemen Konfigurasi VLAN Berbasis VTP dan Layanan DHCP

## Ansible Playbook Implementation to Automate VTP-Based VLAN Configuration Management and DHCP Services

Ni Made Anggrenya Yalestia Chandrawaty<sup>1</sup>, I Putu Hariyadi<sup>2\*</sup>

<sup>1,2</sup>Program Studi Ilmu Komputer Universitas Bumigora

renachandrawati@gmail.com<sup>1</sup>, putu.hariyadi@universitasbumigora.ac.id<sup>2\*</sup>

Submitted: 24 November 2021, Revised: 25 November 2021, Accepted: 13 Desember 2021

**Abstrak** – Penerapan otomatisasi manajemen konfigurasi jaringan dapat meningkatkan efisiensi dan meminimalkan risiko kesalahan konfigurasi. Penelitian terdahulu telah berhasil menerapkan otomatisasi manajemen konfigurasi Virtual Local Area Network (VLAN), InterVLAN dan Dynamic Host Configuration Protocol (DHCP) Server menggunakan Ansible pada infrastruktur jaringan Cisco. Namun rancangan sistem otomatisasi yang dibuat hanya dapat diterapkan pada rancangan jaringan uji coba di penelitian tersebut. Konfigurasi kebijakan masih dikelola secara statik dan tersimpan di file Ansible playbook sehingga ketika terjadi perubahan maka harus mengubah file tersebut. Penelitian ini mengimplementasikan Ansible Playbook dengan variable files yang menampung kebijakan konfigurasi sehingga dapat mengotomatisasi manajemen konfigurasi secara dinamis. Metode penelitian yang digunakan adalah Network Development Life Cycle. Rancangan jaringan uji coba disimulasikan menggunakan PNETLab. Sistem otomatisasi yang dibangun dapat memajemen konfigurasi perangkat jaringan baik router, switch maupun multilayer switch meliputi pembuatan dan penghapusan konfigurasi VLAN berbasis VTP dan DHCP di setiap lokasi. Selain itu sistem otomatisasi yang dibuat dapat mempercepat proses pengelolaan VLAN berdasarkan 3 (tiga) kali percobaan di setiap lokasi yaitu rata-rata 6 (enam) kali lebih cepat ketika operasi pembuatan konfigurasi yaitu lebih cepat 585.3 detik. Sedangkan ketika operasi penghapusan konfigurasi, sistem otomatisasi rata-rata 3 (tiga) kali lebih cepat dibandingkan secara manual yaitu 266.3 detik.

**Kata Kunci:** *ansible, playbook, vlan, vtp, dhcp*

**Abstract** – Implementing network configuration management automation can increase efficiency and minimize the risk of misconfiguration. Previous research has succeeded in implementing automation management of Virtual Local Area Network (VLAN), InterVLAN and Dynamic Host Configuration Protocol (DHCP) Server using Ansible on Cisco network infrastructure. However, the design of the automation system made can only be applied to the trial network design in this study. The policy configuration is still managed statically and stored in the Ansible playbook file so that when a change occurs, it must change the file. This research implements Ansible Playbook with variable files that contain configuration policies so that they can dynamically automate configuration management. The research method used is the Network Development Life Cycle. The trial network design was simulated using PNETLab. The automation system that is built can manage the configuration of network devices, both routers, switches and multilayer switches, including the creation and deletion of VTP and DHCP-based VLAN configurations at each location. In addition, the automation system created can speed up the VLAN management process based on 3 (three) trials at each location, which is an average of 6 (six) times faster when the configuration creation operation is 585.3 seconds faster. Meanwhile, during the configuration deletion operation, the automation system is on average 3 (three) times faster than manually, which is 266.3 seconds.

**Keywords:** *ansible, playbook, vlan, vtp, dhcp*

---

**Jurnal Bumigora Information Technology (BITe)**

Vol.3, No.2, Desember 2021, pp. 107-122

ISSN: 2685-4066

DOI: 10.30812/bite.v3i2.1577

## 1. Pendahuluan

Pertumbuhan bisnis dari suatu perusahaan atau instansi mendorong perluasan pemanfaatan Teknologi Informasi (TI) termasuk infrastruktur jaringan untuk mendukung operasionalnya. Penerapan *Virtual Local Area Network (VLAN)* pada infrastruktur jaringan yang kompleks dapat mempermudah pengontrolan dan pemeliharaan jaringan [1]. VLAN merupakan pengelompokan pengguna atau perangkat secara logikal pada area *broadcast* yang sama di *switch* sehingga mempermudah manajemen dan pengembangan jaringan berdasarkan kebutuhan bisnis dan geografis [2].

Terdapat berbagai penelitian terkait VLAN yang telah dilakukan oleh peneliti sebelumnya. Teknologi VLAN dapat meningkatkan unjuk kerja LAN terkait utilisasi *bandwidth* dengan melokalisasi trafik pengguna dan penerapan *link aggregation* menggunakan *etherchannel* untuk meningkatkan kecepatan jaringan [3]. Teknik *router on a stick* dengan memanfaatkan sebuah *interface* fisik yang secara logikal dipecah menjadi *subinterface* dengan konfigurasi *trunk* dapat digunakan untuk mengkomunikasikan antar VLAN (*InterVLAN*). *Multilayer switch* juga dapat digunakan untuk melakukan *InterVLAN routing* [3], [4]. Proteksi terhadap serangan dapat dilakukan dengan menerapkan *VLAN Access Control List (VACL)* sehingga meningkatkan keandalan jaringan [3].

Penerapan *VLAN Trunking Protocol (VTP)* dapat menyederhanakan administrasi VLAN. VLAN hanya dibuat pada *switch* dengan VTP mode *server*. Sedangkan *switch* lainnya diatur sebagai VTP mode *client*. VTP akan secara otomatis mengirimkan pembaruan terkait VLAN ke *switch* lainnya yang berada pada *domain* yang sama [3], [4]. Penelitian terdahulu hanya berfokus pada manajemen VLAN dan VTP menggunakan konfigurasi secara manual. Pengelolaan infrastruktur jaringan menggunakan pendekatan manual dapat mengakibatkan biaya operasional meningkat, tidak efisien dan rentan terhadap kesalahan seiring dengan bertambahnya perangkat jaringan yang dikelola.

Penelitian [5] telah berhasil menerapkan otomatisasi manajemen konfigurasi *Virtual Local Area Network (VLAN)*, *InterVLAN* dan *Dynamic Host Configuration Protocol (DHCP) Server* menggunakan *Ansible* pada infrastruktur jaringan berbasis *Cisco*. *Ansible* merupakan tool bersifat *open source* dan otomatisasi *agentless* yang dapat digunakan untuk melakukan manajemen konfigurasi, penyebaran dan orkestrasi TI termasuk *cloud provisioning* [6][7]. Sedangkan DHCP merupakan metode alokasi pengalamanan IP secara dinamis ke *client* [8]. Sistem otomatisasi yang dikembangkan tersebut rata-rata dapat mempercepat proses konfigurasi sebanyak 3 (tiga) kali yang diterapkan pada rancangan jaringan uji coba dengan 4 (empat) *router* dan *switch* yaitu 25 menit 24 detik dibandingkan secara manual dengan waktu 1 jam 21 menit [5]. Pendekatan pengelolaan infrastruktur jaringan menggunakan otomatisasi dapat meminimalkan risiko perubahan pada jaringan produksi yang dapat mengakibatkan pelayanan menurun. Selain itu mengurangi pekerjaan berulang dan meningkatkan efisiensi karena otomatisasi jaringan dapat mengelola jaringan dengan cepat.

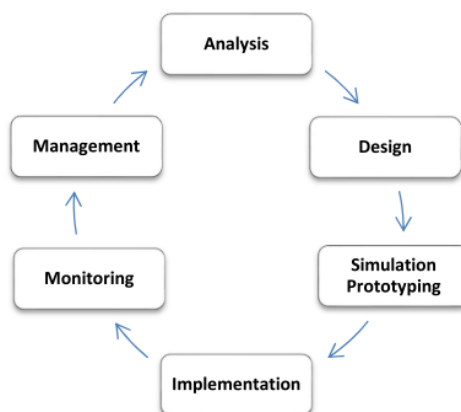
Kelemahan dari hasil penelitian [5] adalah kebijakan konfigurasi pada rancangan sistem otomatisasi yang digunakan masih dikelola secara statik dan tersimpan pada *file Ansible Playbook*. Selain itu rancangan sistem otomatisasinya hanya dapat diterapkan pada topologi yang telah dibuat sesuai rancangan jaringan uji coba pada penelitian tersebut. Dampaknya ketika terjadi suatu perubahan topologi jaringan dan kebijakan konfigurasi maka penyesuaian harus dilakukan dengan mengubah *file Ansible playbook* secara langsung sehingga tidak fleksibel.

Mendorong ketertarikan penulis untuk melakukan penelitian terkait rancangan sistem otomatisasi yang mengakomodir kebijakan konfigurasi infrastruktur jaringan secara dinamis sehingga mengurangi kompleksitas pengelolaan. Selain itu menerapkan rancangan sistem tersebut menggunakan *tool* otomatisasi jaringan. Penelitian ini menggunakan metode *variable files* dengan format *YAML* untuk menampung kebijakan konfigurasi infrastruktur jaringan yang diterapkan untuk lokasi yang berbeda. *Ansible Playbook* memanggil *variable files* untuk dieksekusi berdasarkan pada parameter lokasi sehingga dapat mengotomatisasi manajemen konfigurasi infrastruktur jaringan secara dinamis. *Ansible playbook* merupakan *file* yang

memuat serangkaian tugas atau *tasks* yang memicu *module Ansible* tertentu dan ditulis menggunakan format YAML [9]. Dengan adanya penelitian ini diharapkan dapat memberikan wawasan, pengetahuan, serta solusi dalam manajemen konfigurasi infrastruktur jaringan secara dinamis sehingga lebih efektif, efisien dan terstruktur.

## 2. Metode Penelitian

Metode penelitian yang digunakan adalah *Network Development Life Cycle (NDLC)*. NDLC terdiri dari 6 (enam) tahapan yaitu *Analysis*, *Design*, *Simulation Prototyping*, *Implementation*, *Monitoring* dan *Management*, seperti terlihat pada gambar 1 [10]. Peneliti hanya menggunakan 3 (tiga) dari 6 (enam) tahapan yang terdapat pada NDLC yaitu *Analysis*, *Design*, dan *Simulation Prototyping*.



Gambar 1. Tahapan Network Development Life Cycle [10]

### 2.1. Tahap Analysis

Pada tahap ini dilakukan pengumpulan data menggunakan studi literatur dan analisa terhadap data yang telah dikumpulkan. Pengumpulan data berupa artikel ilmiah terkait manajemen konfigurasi VLAN dan DHCP pada infrastruktur jaringan *Cisco*. Berdasarkan hasil analisa terhadap data yang telah dikumpulkan tersebut maka diperoleh informasi bahwa penelitian sebelumnya berfokus pada manajemen konfigurasi VLAN dan DHCP secara manual. Selain itu terdapat pula penelitian yang telah menerapkan otomatisasi manajemen konfigurasi VLAN dan DHCP menggunakan *Ansible* namun kebijakan konfigurasi masih secara statik dan belum menerapkan VTP untuk mensinkronisasi informasi terkait VLAN antar *switch*. Mendorong ketertarikan peneliti untuk menindaklanjuti implementasi *Ansible Playbook* untuk mengotomatisasi manajemen konfigurasi VLAN Berbasis VTP Dan Layanan DHCP secara dinamis.

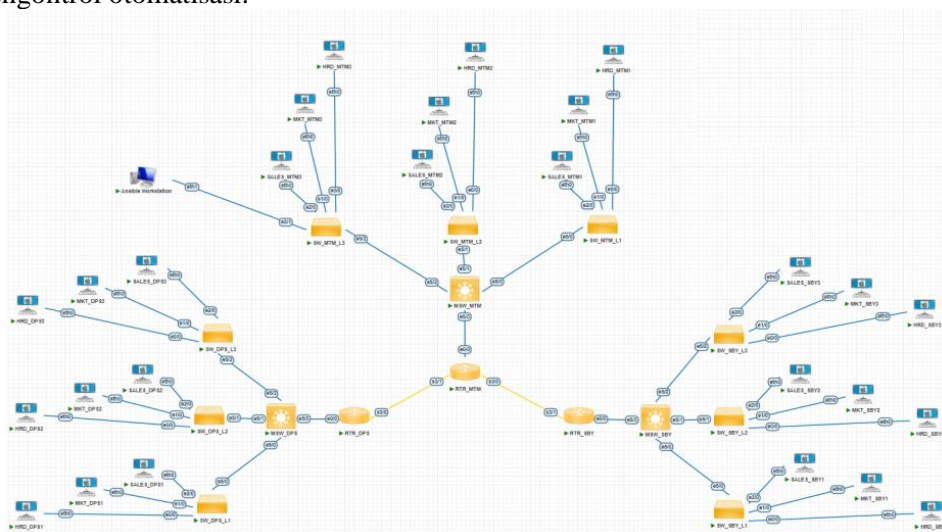
### 2.2. Tahap Design

Tahap ini terdiri dari rancangan jaringan ujicoba, rancangan pengalamatan IP dan rancangan sistem otomatisasi.

#### 2.2.1. Rancangan Jaringan Ujicoba

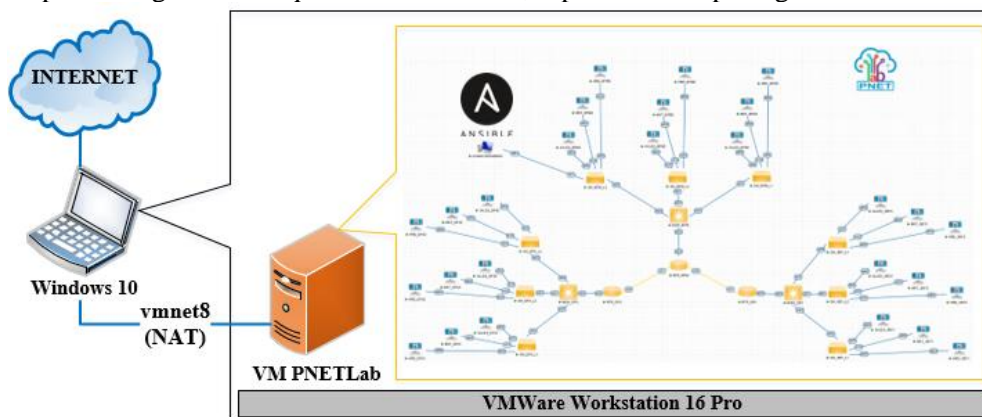
Adapun rancangan jaringan ujicoba yang digunakan pada penelitian ini, seperti terlihat pada gambar 2. Terlihat sebuah rancangan jaringan ujicoba dari suatu perusahaan yang memiliki kantor pusat di Mataram dan dua kantor cabang yaitu berlokasi di Denpasar dan Surabaya. Pada kantor pusat memiliki empat jaringan yaitu *Server*, *Human Resources Development (HRD)*, *Marketing* dan *Sales*. Sedangkan pada setiap kantor cabang terdapat tiga jaringan yaitu HRD, *Marketing* dan *Sales*. Setiap lokasi menggunakan perangkat jaringan berupa satu unit *router*, satu unit *multilayer switch* dan tiga unit *switch*, serta sembilan buah *Personal Computer (PC)* yang tersebar di tiga lantai dari gedung perusahaan. Namun khusus untuk lokasi kantor pusat

Mataram, terdapat satu buah perangkat tambahan yaitu *server Ansible* yang bertindak sebagai *node* pengontrol otomatisasi.



Gambar 2. Rancangan Jaringan Ujicoba

Rancangan jaringan ujicoba tersebut disimulasikan menggunakan *Packet Network Emulator Tool Lab (PNETLab)*. *PNETLab* merupakan platform yang dapat digunakan untuk membuat, berbagi dan mempraktekkan laboratorium jaringan yang mendukung beragam vendor. *PNETLab* diinstalasi pada *hypervisor VMWare Workstation* yang diinstalasi pada satu unit komputer dengan *sistem operasi Windows 10*, seperti terlihat pada gambar 3.



Gambar 3. Simulasi Rancangan Jaringan Ujicoba dengan PNETLab

2.2.2. Rancangan Pengalamanan IP

Rancangan pengalamanan IP yang digunakan pada jaringan ujicoba adalah alamat *network class B 172.31.0.0/16* yang di *subnetting*. Detail alokasi pengalamanan IP yang dialokasikan untuk setiap perangkat di masing-masing lokasi, seperti terlihat pada tabel 1.

Tabel 1. Rancangan Pengalamanan IP

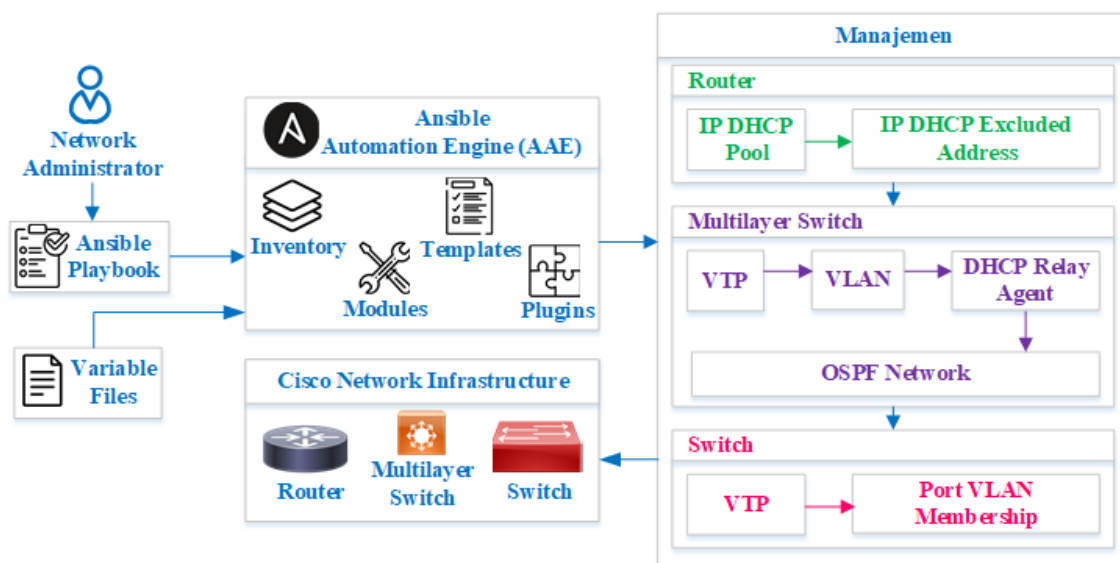
No.	Nama Perangkat	Interface	IP	Gateway	Deskripsi
1.	RTR_MTM	E0/0	172.31.0.1/27		Router di kantor pusat Mataram (MTM) yang terhubung ke <i>Multilayer Switch MSW_MTM</i>
		S3/1	172.31.3.1/27		Terhubung ke <i>router Denpasar (RTR_DPS)</i>

		S3/0	172.31.4.1/27		Terhubung ke router Surabaya (RTR_SBY)
2.	MSW_MTM	E5/3	172.31.0.2/27		Terhubung ke RTR_MTRM
		VLAN 1	172.31.0.130/27		IP untuk manajemen switch.
		VLAN 2	172.31.0.33/27		Terhubung ke VLAN HRD MTM
		VLAN 3	172.31.0.65/27		Terhubung ke VLAN MKT MTM
		VLAN 4	172.31.0.97/27		Terhubung ke VLAN SALES MTM
3.	SW_MTM_L1	VLAN 1	172.31.0.131/27	172.31.0.130/27	IP untuk manajemen switch.
4.	SW_MTM_L2	VLAN 1	172.31.0.132/27		
5.	SW_MTM_L3	VLAN 1	172.31.0.133/27		
6.	RTR_DPS	E0/0	172.31.1.1/27		Terhubung ke MSW_DPS
		S3/0	172.31.3.2/27		Terhubung ke RTR_MTM
7.	MSW_DPS	E5/3	172.31.1.2/27		Terhubung ke RTR_DPS
		VLAN 1	172.31.1.130/27		IP untuk manajemen switch.
		VLAN 2	172.31.33/27		Terhubung ke VLAN HRD DPS
		VLAN 3	172.31.1.65/27		Terhubung ke VLAN MKT DPS
		VLAN 4	172.31.1.97/27		Terhubung ke VLAN SALES DPS
8.	SW_DPS_L1	VLAN 1	172.31.1.131/27	172.31.1.130/27	IP untuk manajemen switch.
9.	SW_DPS_L2	VLAN 1	172.31.1.132/27		
10.	SW_DPS_L3	VLAN 1	172.31.1.133/27		
11.	RTR_SBY	E0/0	172.31.2.1/27		Terhubung ke MSW_SBY
		S3/1	172.31.4.2/27		Terhubung ke RTR_MTRM
12.	MSW_SBY	E5/3	172.31.2.2/27		Terhubung ke RTR_SBY
		VLAN 1	172.31.2.130/27		IP untuk manajemen switch.
		VLAN 2	172.31.2.33/27		Terhubung ke VLAN HRD SBY

		VLAN 3	172.31.2.65/27		Terhubung ke VLAN MKT SBY
		VLAN 4	172.31.2.97/27		Terhubung ke VLAN SALES SBY
13.	SW_SBY_L1	VLAN 1	172.31.2.131/27	172.31.2.130/27	IP untuk manajemen switch.
14.	SW_SBY_L2	VLAN 1	172.31.2.132/27		
15.	SW_SBY_L3	VLAN 1	172.31.2.133/27		

**2.2.3. Rancangan Sistem Otomatisasi**

Adapun rancangan sistem otomatisasi yang dibuat pada penelitian ini, seperti terlihat pada gambar 4.



**Gambar 4. Rancangan Sistem Otomatisasi**

Network Administrator membuat Ansible Playbook yang didalamnya memuat definisi tugas (tasks) otomatisasi yang ditulis menggunakan format YAML. Selain itu juga membuat variable files yang didalamnya memuat deklarasi variable terkait kebijakan manajemen konfigurasi secara dinamis yang diterapkan pada setiap perangkat jaringan di masing-masing lokasi. Kebijakan manajemen konfigurasi pada router meliputi IP DHCP Pool dan IP DHCP Excluded Address. Kebijakan manajemen konfigurasi yang dilakukan pada multilayer switch meliputi manajemen VTP, VLAN, DHCP Relay Agent dan OSPF Network. Sedangkan kebijakan manajemen konfigurasi yang dilakukan pada switch meliputi VTP dan Port VLAN Membership.

Ansible playbook saat dieksekusi akan berinteraksi dengan Ansible Automation Engine (AAE) yang didalamnya memuat inventory, modules, templates dan plugins. Inventory merupakan file inisialisasi yang memuat daftar perangkat infrastruktur jaringan Cisco yang dikelola meliputi router, multilayer switch dan switch dari ketiga lokasi. Sedangkan modules ansible utama yang digunakan adalah ios\_config yang berfungsi untuk memajemen bagian konfigurasi dari Cisco IOS pada perangkat jaringan yang dikelola.

**2.3. Tahap Simulation Prototyping**

Tahap ini terdiri dari instalasi dan konfigurasi, pembuatan Ansible Inventory dan Playbook serta ujicoba. Instalasi dan konfigurasi meliputi melakukan import PNETLab pada VMWare

Workstation, dan pembuatan topologi berdasarkan rancangan jaringan uji coba pada *PNETLab*, serta konfigurasi dasar pada setiap perangkat jaringan pada topologi. Konfigurasi dasar pada topologi *PNETLab* dilakukan pada 3 (tiga) *router* dan *multilayer switch*, 9 (sembilan) *switch*, 27 (duapuluh tujuh) *PC* dan satu *server Ansible*. Sedangkan pada bagian pembuatan *Ansible playbook* dilakukan penyusunan kode program sesuai dengan rancangan sistem otomatisasi sehingga dapat menerapkan kebijakan manajemen konfigurasi VLAN berbasis VTP dan layanan DHCP secara dinamis untuk setiap lokasi.

Uji coba dibagi menjadi 2 (dua) yaitu verifikasi konfigurasi dan skenario. Verifikasi konfigurasi berupa pemeriksaan konfigurasi dasar dari setiap perangkat jaringan. Sedangkan skenario berupa pembuatan dan penghapusan konfigurasi secara manual dan otomatisasi menggunakan *Ansible Playbook*. Uji coba berbasis skenario tersebut dilakukan sebanyak 3 (tiga) kali pada setiap perangkat jaringan meliputi *router*, *multilayer switch* dan *switch* yang terdapat di setiap lokasi. Selain itu juga dilakukan perhitungan waktu untuk setiap skenario pengujian tersebut sehingga diketahui selisih waktu antara manajemen konfigurasi manual dengan otomatisasi meliputi waktu minimum, maksimum dan rata-rata.

### 3. Hasil dan Pembahasan

Pada bagian ini membahas tentang hasil instalasi dan konfigurasi, *Ansible playbook*, uji coba serta analisa terhadap hasil uji coba.

#### 3.1. Hasil Instalasi dan Konfigurasi

Terdapat instalasi dan konfigurasi yang dilakukan meliputi *import PNETLab* sebagai *Virtual Machine (VM)* dan membuat topologi melalui antarmuka berbasis web dari *PNETLab* sesuai dengan rancangan topologi jaringan uji coba, serta konfigurasi dasar setiap perangkat jaringan. *PNETLab* yang telah berhasil di *import* dan dijalankan sebagai VM di *VMWare Workstation*, terlihat seperti pada gambar 5.

```
PNETLab (default root password is 'pnet')
Use https or http://192.168.196.131/

WARNING: neither Intel VT-x or AMD-V found

pnetlab login: _
```

Gambar 5. Hasil Instalasi PNETLab sebagai VM

#### 3.2. Hasil Ansible Inventory, Variable Files dan Playbook

*Ansible inventory* memuat pengelompokkan perangkat jaringan yang dimanajemen berdasarkan lokasi dan jenis perangkat, seperti terlihat pada gambar 6.

```
[rtr_matarani]
RTR_MTM ansible_host=172.31.0.1 ansible_user=admin ansible_ssh_pass=anggreana ansible_connection=network_cli ansible_network_os=ios
[msw_matarani]
MSW_MTM ansible_host=172.31.0.130 ansible_user=admin ansible_ssh_pass=anggreana ansible_connection=network_cli ansible_network_os=ios

[sw_matarani]
SW_MTM_L1 ansible_host=172.31.0.131 ansible_user=admin ansible_ssh_pass=anggreana ansible_connection=network_cli ansible_network_os=ios
SW_MTM_L2 ansible_host=172.31.0.132 ansible_user=admin ansible_ssh_pass=anggreana ansible_connection=network_cli ansible_network_os=ios
SW_MTM_L3 ansible_host=172.31.0.133 ansible_user=admin ansible_ssh_pass=anggreana ansible_connection=network_cli ansible_network_os=ios

[rtr_denpasar]
RTR_DPS ansible_host=172.31.1.1 ansible_user=admin ansible_ssh_pass=anggreana ansible_connection=network_cli ansible_network_os=ios
[msw_denpasar]
MSW_DPS ansible_host=172.31.1.130 ansible_user=admin ansible_ssh_pass=anggreana ansible_connection=network_cli ansible_network_os=ios

[sw_denpasar]
SW_DPS_L1 ansible_host=172.31.1.131 ansible_user=admin ansible_ssh_pass=anggreana ansible_connection=network_cli ansible_network_os=ios
SW_DPS_L2 ansible_host=172.31.1.132 ansible_user=admin ansible_ssh_pass=anggreana ansible_connection=network_cli ansible_network_os=ios
SW_DPS_L3 ansible_host=172.31.1.133 ansible_user=admin ansible_ssh_pass=anggreana ansible_connection=network_cli ansible_network_os=ios

[rtr_surabaya]
RTR_SBY ansible_host=172.31.2.1 ansible_user=admin ansible_ssh_pass=anggreana ansible_connection=network_cli ansible_network_os=ios
[msw_surabaya]
MSW_SBY ansible_host=172.31.2.130 ansible_user=admin ansible_ssh_pass=anggreana ansible_connection=network_cli ansible_network_os=ios

[sw_surabaya]
SW_SBY_L1 ansible_host=172.31.2.131 ansible_user=admin ansible_ssh_pass=anggreana ansible_connection=network_cli ansible_network_os=ios
SW_SBY_L2 ansible_host=172.31.2.132 ansible_user=admin ansible_ssh_pass=anggreana ansible_connection=network_cli ansible_network_os=ios
SW_SBY_L3 ansible_host=172.31.2.133 ansible_user=admin ansible_ssh_pass=anggreana ansible_connection=network_cli ansible_network_os=ios
```

Gambar 6. Ansible Inventory

Terdapat 5 (lima) variable yang diatur pada setiap perangkat jaringan yang dimanajemen meliputi *ansible\_host*, *ansible\_ssh\_pass*, *ansible\_connect* dan *ansible\_network* serta

*ansible\_network\_os*. *ansible\_host* merupakan *variable* untuk menentukan alamat IP dari perangkat jaringan yang dimanajemen yaitu *router*, *multilayer switch* dan *switch* di lokasi Mataram, Denpasar dan Surabaya. *ansible\_user* merupakan *variable* untuk menentukan *user* yang digunakan untuk terkoneksi ke perangkat jaringan yaitu “admin”. *ansible\_ssh\_pass* merupakan *variable* untuk menentukan sandi *login* dari *user* yang digunakan untuk terkoneksi ke perangkat jaringan yaitu “anggrenas”. *ansible\_connection* merupakan *variable* untuk mengatur agar *Ansible* memperlakukan perangkat yang dimanajemen sebagai perangkat jaringan dengan lingkungan eksekusi yang terbatas yaitu “network\_cli”. Sedangkan *ansible\_network\_os* merupakan *variable* untuk menginformasikan platform jaringan dari perangkat yang dimanajemen yaitu “ios”.

*Ansible variable files* memuat deklarasi nilai dari konfigurasi VLAN berbasis VTP dan layanan DHCP untuk setiap lokasi dalam format YAML. Terdapat 3 (tiga) *variable files* yaitu *var\_mataram.yml* untuk lokasi Mataram, *var\_denpasar.yml* untuk lokasi Denpasar dan *var\_surabaya.yml* untuk lokasi Surabaya. Sebagai salah satu contoh terkait isi dari *variable file* *var\_mataram.yml* untuk lokasi Mataram, seperti terlihat pada gambar 7.

```

1  dhcp:
2    - { departemen: "HRD_MTM", network: "172.31.0.32", mask: "255.255.255.224", default_router:
      "172.31.0.33" }
3    - { departemen: "MKT_MTM", network: "172.31.0.64", mask: "255.255.255.224", default_router:
      "172.31.0.65" }
4    - { departemen: "SALES_MTM", network: "172.31.0.96", mask: "255.255.255.224", default_router:
      "172.31.0.97" }
5
6  vtp:
7    - { name: "MTM" }
8
9  vlan_dan_ospf:
10   - { vlanid: 2, departemen: "HRD", name: "MTM", network_interface: "172.31.0.33", mask_interface:
      "255.255.255.224", helper_address: "172.31.0.1", network_ospf: "172.31.0.32", mask_ospf: "0.0.0.31" }
11   - { vlanid: 3, departemen: "MKT", name: "MTM", network_interface: "172.31.0.65", mask_interface:
      "255.255.255.224", helper_address: "172.31.0.1", network_ospf: "172.31.0.64", mask_ospf: "0.0.0.31" }
12   - { vlanid: 4, departemen: "SALES", name: "MTM", network_interface: "172.31.0.97", mask_interface:
      "255.255.255.224", helper_address: "172.31.0.1", network_ospf: "172.31.0.96", mask_ospf: "0.0.0.31" }
13
14  portmembership:
15   - { vlanid: 2, name: "MTM", portmembership: "et0/0-3" }
16   - { vlanid: 3, name: "MTM", portmembership: "et1/0-3" }
17   - { vlanid: 4, name: "MTM", portmembership: "et2/0-3" }

```

**Gambar 7 Ansible Variable Files**

Terlihat terdapat 4 (empat) *array dictionary* dengan nama *dhcp*, *vtp*, *vlan\_dan\_ospf* dan *portmembership*. *Dictionary dhcp* didalamnya memuat pasangan *key* dan *value* yang diperlukan untuk konfigurasi 3 (tiga) IP DHCP *pool* sesuai dengan *value* dari *key* departemen meliputi HRD\_MTM, MKT\_MTM dan SALES\_MTM. Setiap *pool* memiliki pengaturan parameter *network* dan *subnetmask* serta *default gateway* dengan *value* yang diambil masing-masing dari *key network*, *mask* dan *default\_router*. IP DHCP *Pool* HRD\_MTM memiliki *network* 172.31.0.32 dengan *subnetmask* 255.255.255.224 dan *default-router* 172.31.0.33. IP DHCP *Pool* MKT\_MTM memiliki *network* 172.31.0.64 dengan *subnetmask* 255.255.255.224 dan *default-router* 172.31.0.65. Sedangkan IP DHCP *Pool* SALES\_MTM memiliki *network* 172.31.0.96 dengan *subnetmask* 255.255.255.224 dan *default-router* 172.31.0.97.

*Dictionary vtp* didalamnya memuat pasangan *key* dan *value* untuk menentukan nama domain VTP yaitu MTM. *Dictionary vlan\_dan\_ospf* didalamnya memuat pasangan *key* dan *value* untuk pembuatan VLAN, alokasi pengalamatan IP untuk setiap *interface* VLAN, DHCP *Relay Agent* dengan nilai 172.31.0.1 serta OSPF *network parameter* untuk *subnet* setiap VLAN. Terdapat 3 (tiga) VLAN yang dibuat yaitu ID 2 dengan nama HRD, ID 3 dengan nama MKT, ID 4 dengan nama SALES. Setiap *interface* dari VLAN memiliki pengaturan alamat IP yaitu VLAN ID 2 menggunakan 172.31.0.33/27, VLAN ID 3 menggunakan 172.31.0.65/27 dan VLAN ID 4 menggunakan 172.31.0.97/27. Selain itu OSPF *network parameter* yang dikonfigurasi meliputi 172.31.0.32, 172.31.0.64 dan 172.31.0.96 dengan *wildcard mask* 0.0.0.31. Terakhir *dictionary portmembership* didalamnya memuat pasangan *key* dan *value* untuk keanggotaan *interface* untuk setiap VLAN. VLAN ID 2 dengan *interface* anggota et0/0-3, VLAN ID 3 dengan *interface* anggota et1/0-3 dan VLAN ID 4 dengan *interface* anggota et2/0-3.



Terdapat 11 (sebelas) *file Ansible playbook* yang dihasilkan untuk otomatisasi pembuatan konfigurasi VLAN berbasis VTP dan layanan DHCP yaitu *main.yml*, *router.yml*, *msw.yml*, *sw.yml*, *task\_router.yml*, *task\_msw.yml* dan *task\_sw.yml*. Sedangkan untuk otomatisasi penghapusan konfigurasi terdapat 4 (empat) *file* yaitu *delete\_main.yml*, *delete\_config\_router.yml* dan *delete\_config\_msw.yml* serta *delete\_config\_sw.yml*. Adapun isi dari *main.yml* yang merupakan *file playbook* utama untuk mengeksekusi otomatisasi pembuatan konfigurasi, seperti terlihat pada gambar 8.

```
1 ---
2 - name: Main
3   hosts: localhost
4   connection: network_cli
5   gather_facts: no
6   become: yes
7   become_method: enable
8   ignore_errors: yes
9   tasks:
10  - name: include variable
11    include_vars: var_{{ lokasi }}.yaml
12
13  - name: Variable untuk Router di Setiap Lokasi
14    set_fact:
15      router: "rtr_{{ lokasi }}"
16
17  - name: Task untuk Router
18    include_tasks: router.yml
19    with_items: "{{ groups[router] }}"
20
21  - name: Variable untuk Multilayer Switch di Setiap Lokasi
22    set_fact:
23      msw: "msw_{{ lokasi }}"
24
25  - name: Task untuk Multilayer Switch
26    include_tasks: msw.yml
27    with_items: "{{ groups[msw] }}"
28
29  - name: Variable untuk Switch di Setiap Lokasi
30    set_fact:
31      switch: "sw_{{ lokasi }}"
32
33  - name: Task untuk Switch
34    include_tasks: sw.yml
35    with_items: "{{ groups[switch] }}"
```

Gambar 8. Ansible Playbook

Baris 1 merupakan awal dari dokumen YAML yang ditandai dengan 3 (*three*) *hyphen*. Baris 2 digunakan untuk menentukan nama dari *ansible playbook* yaitu "Main". Baris 3 digunakan untuk menentukan daftar *host* atau *group host* sebagai lokasi eksekusi *task* yaitu *localhost*. Baris 4 digunakan untuk mengatur agar *Ansible* memperlakukan perangkat yang dimanajemen sebagai perangkat jaringan yaitu *network\_cli*. Baris 5 digunakan untuk menonaktifkan pengumpulan informasi terkait *remote host*. Baris 6 dan 7 digunakan mengaktifkan *privilege escalation* dan memastikan *Ansible* dapat menggunakan hak tersebut. Baris 8 digunakan agar *playbook* tetap berjalan ketika terjadi kesalahan pada *task* tertentu.

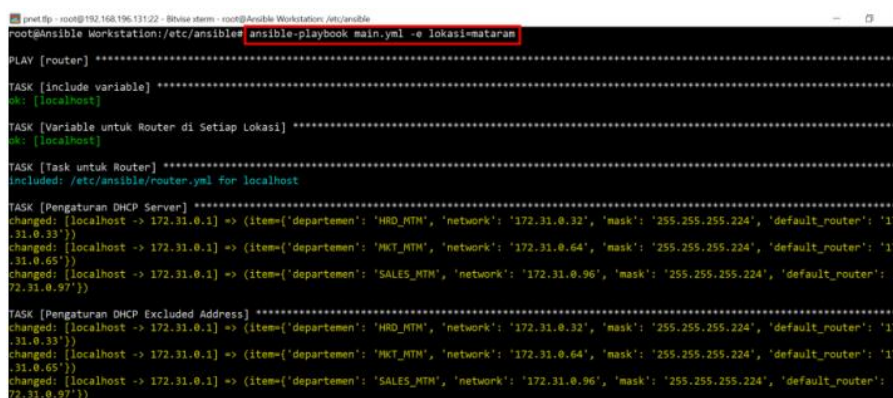
Baris 9 digunakan untuk mendeklarasikan *tasks*. Baris 10 dan 11 digunakan untuk mendeklarasikan *task* dengan nama "include variable" dan menyisipkan *file variable* menggunakan modul *include\_vars* yaitu dengan awalan *var\_* yang diikuti dengan lokasi dan diakhir dengan ekstensi *.yaml*. Baris ke 13 sampai dengan 15 merupakan deklarasi *task* dengan nama "Variable untuk Router di Setiap Lokasi" dan menggunakan *module set\_fact* untuk mengatur *variable* baru dengan nama "router" dengan nilai berupa *group host* dari *router* yaitu dengan awalan *rtr\_* dan diikuti dengan lokasi. Baris ke 17 sampai dengan 19 merupakan deklarasi *task* dengan nama "Task untuk Router" dan menyisipkan *file task router.yml* menggunakan *module include\_tasks*. *File task router.yml* memuat tugas untuk pengaturan DHCP Server dan DHCP *excluded address* yang terdapat di lokasi tertentu.

Baris ke 21 sampai dengan 23 merupakan deklarasi *task* dengan nama "Variable untuk Multilayer Switch di Setiap Lokasi" dan menggunakan *module set\_fact* untuk mengatur *variable* baru dengan nama "msw" dengan nilai berupa *group host* dari *multilayer switch* yaitu dengan awalan *msw\_* dan diikuti dengan lokasi. Baris ke 25 sampai dengan 27 merupakan deklarasi *task* dengan nama "Task untuk Multilayer Switch" dan menyisipkan *file task msw.yml* menggunakan *module include\_tasks*. *File task msw.yml* memuat tugas untuk pengaturan mode VTP, nama

*domain* VTP, pembuatan VLAN, pengaturan IP untuk setiap *interface* dari VLAN dan *DHCP Relay Agent* serta *OSPF network parameter* untuk setiap VLAN. Baris ke 29 sampai dengan 31 merupakan deklarasi *task* dengan nama "Variable untuk Switch di Setiap Lokasi" dan menggunakan *module set\_fact* untuk mengatur *variable* baru dengan nama "switch" dengan nilai berupa *group host* dari *switch* yaitu dengan awalan *sw\_* dan diikuti dengan lokasi. Baris ke 33 sampai dengan 35 merupakan deklarasi *task* dengan nama "Task untuk Switch" dan menyisipkan *file task sw.yml* menggunakan *module include\_tasks*. *File task sw.yml* memuat tugas untuk pengaturan mode VTP dan nama *domain* VTP serta *port membership* setiap VLAN yang operasinya diulang sejumlah *switch* yang terdapat di lokasi tertentu.

### 3.3. Hasil Ujicoba

Percobaan pembuatan dan penghapusan konfigurasi VLAN berbasis VTP dan DHCP baik secara manual maupun otomatisasi dilakukan sebanyak 3 (tiga) kali di masing-masing lokasi yaitu Mataram, Denpasar, dan Surabaya. Gambar 9 memperlihatkan cuplikan hasil eksekusi *playbook* untuk mengotomatisasi pembuatan konfigurasi VLAN berbasis VTP dan DHCP di lokasi Mataram yang dipicu menggunakan perintah `ansible-playbook main.yml -e lokasi=mataram`.



```
root@Ansible Workstation: /etc/ansible # ansible-playbook main.yml -e lokasi=mataram
PLAY [router] *****
TASK [include variable] *****
ok: [localhost]

TASK [Variable untuk Router di Setiap Lokasi] *****
ok: [localhost]

TASK [Task untuk Router] *****
included: /etc/ansible/router.yml for localhost

TASK [Pengaturan DHCP Server] *****
changed: [localhost -> 172.31.0.1] => (item={'departemen': 'HRD_MTM', 'network': '172.31.0.32', 'mask': '255.255.255.224', 'default_router': '172.31.0.33'})
changed: [localhost -> 172.31.0.1] => (item={'departemen': 'MKT_MTM', 'network': '172.31.0.64', 'mask': '255.255.255.224', 'default_router': '172.31.0.65'})
changed: [localhost -> 172.31.0.1] => (item={'departemen': 'SALES_MTM', 'network': '172.31.0.96', 'mask': '255.255.255.224', 'default_router': '172.31.0.97'})

TASK [Pengaturan DHCP Excluded Address] *****
changed: [localhost -> 172.31.0.1] => (item={'departemen': 'HRD_MTM', 'network': '172.31.0.32', 'mask': '255.255.255.224', 'default_router': '172.31.0.33'})
changed: [localhost -> 172.31.0.1] => (item={'departemen': 'MKT_MTM', 'network': '172.31.0.64', 'mask': '255.255.255.224', 'default_router': '172.31.0.65'})
changed: [localhost -> 172.31.0.1] => (item={'departemen': 'SALES_MTM', 'network': '172.31.0.96', 'mask': '255.255.255.224', 'default_router': '172.31.0.97'})
```

Gambar 9. Hasil Eksekusi Playbook untuk Task Membuat Konfigurasi di Router Lokasi Mataram

Terlihat terdapat 5 (lima) *tasks* yang dieksekusi yaitu *include variable*, *task* untuk *router*, pengaturan *DHCP Server* berupa dan pengaturan *DHCP Excluded Address*. Sedangkan cuplikan hasil pembuatan konfigurasi di perangkat *multilayer switch* untuk lokasi Mataram, seperti terlihat pada gambar 10.

```
TASK [Variable untuk Multilayer Switch di Setiap Lokasi] *****
ok: [localhost]

TASK [Task untuk Multilayer Switch] *****
included: /etc/ansible/mw.yml for localhost

TASK [Pengaturan Mode VTP dan VTP Domain Name] *****
changed: [localhost -> 172.31.0.130] => (item={'name': 'MTH'})

TASK [Pembuatan VLAN] *****
changed: [localhost -> 172.31.0.130] => (item={'vlanid': 2, 'departemen': 'HRD', 'name': 'MTH', 'network_interface': '172.31.0.33', 'mask_interface': '255.255.255.224', 'helper_address': '172.31.0.1', 'network_ospf': '172.31.0.32', 'mask_ospf': '0.0.0.31'})
changed: [localhost -> 172.31.0.130] => (item={'vlanid': 3, 'departemen': 'MKT', 'name': 'MTH', 'network_interface': '172.31.0.65', 'mask_interface': '255.255.255.224', 'helper_address': '172.31.0.1', 'network_ospf': '172.31.0.64', 'mask_ospf': '0.0.0.31'})
changed: [localhost -> 172.31.0.130] => (item={'vlanid': 4, 'departemen': 'SALES', 'name': 'MTH', 'network_interface': '172.31.0.97', 'mask_interface': '255.255.255.224', 'helper_address': '172.31.0.1', 'network_ospf': '172.31.0.96', 'mask_ospf': '0.0.0.31'})

TASK [Pengaturan IP per interface VLAN] *****
changed: [localhost -> 172.31.0.130] => (item={'vlanid': 2, 'departemen': 'HRD', 'name': 'MTH', 'network_interface': '172.31.0.33', 'mask_interface': '255.255.255.224', 'helper_address': '172.31.0.1', 'network_ospf': '172.31.0.32', 'mask_ospf': '0.0.0.31'})
changed: [localhost -> 172.31.0.130] => (item={'vlanid': 3, 'departemen': 'MKT', 'name': 'MTH', 'network_interface': '172.31.0.65', 'mask_interface': '255.255.255.224', 'helper_address': '172.31.0.1', 'network_ospf': '172.31.0.64', 'mask_ospf': '0.0.0.31'})
changed: [localhost -> 172.31.0.130] => (item={'vlanid': 4, 'departemen': 'SALES', 'name': 'MTH', 'network_interface': '172.31.0.97', 'mask_interface': '255.255.255.224', 'helper_address': '172.31.0.1', 'network_ospf': '172.31.0.96', 'mask_ospf': '0.0.0.31'})

TASK [Pengaturan DHCP Relay Agent] *****
changed: [localhost -> 172.31.0.130] => (item={'vlanid': 2, 'departemen': 'HRD', 'name': 'MTH', 'network_interface': '172.31.0.33', 'mask_interface': '255.255.255.224', 'helper_address': '172.31.0.1', 'network_ospf': '172.31.0.32', 'mask_ospf': '0.0.0.31'})
changed: [localhost -> 172.31.0.130] => (item={'vlanid': 3, 'departemen': 'MKT', 'name': 'MTH', 'network_interface': '172.31.0.65', 'mask_interface': '255.255.255.224', 'helper_address': '172.31.0.1', 'network_ospf': '172.31.0.64', 'mask_ospf': '0.0.0.31'})
changed: [localhost -> 172.31.0.130] => (item={'vlanid': 4, 'departemen': 'SALES', 'name': 'MTH', 'network_interface': '172.31.0.97', 'mask_interface': '255.255.255.224', 'helper_address': '172.31.0.1', 'network_ospf': '172.31.0.96', 'mask_ospf': '0.0.0.31'})

TASK [Penambahan Network Parameter untuk Setiap VLAN] *****
changed: [localhost -> 172.31.0.130] => (item={'vlanid': 2, 'departemen': 'HRD', 'name': 'MTH', 'network_interface': '172.31.0.33', 'mask_interface': '255.255.255.224', 'helper_address': '172.31.0.1', 'network_ospf': '172.31.0.32', 'mask_ospf': '0.0.0.31'})
changed: [localhost -> 172.31.0.130] => (item={'vlanid': 3, 'departemen': 'MKT', 'name': 'MTH', 'network_interface': '172.31.0.65', 'mask_interface': '255.255.255.224', 'helper_address': '172.31.0.1', 'network_ospf': '172.31.0.64', 'mask_ospf': '0.0.0.31'})
changed: [localhost -> 172.31.0.130] => (item={'vlanid': 4, 'departemen': 'SALES', 'name': 'MTH', 'network_interface': '172.31.0.97', 'mask_interface': '255.255.255.224', 'helper_address': '172.31.0.1', 'network_ospf': '172.31.0.96', 'mask_ospf': '0.0.0.31'})
```

Gambar 10. Hasil Eksekusi Playbook untuk Task Membuat Konfigurasi di Multilayer Switch Lokasi Mataram

Terlihat terdapat 7 (tujuh) *tasks* yang dieksekusi yaitu *variable* untuk *multilayer switch* di setiap lokasi, *task* untuk *multilayer switch*, pengaturan mode VTP dan VTP *Domain Name*, pembuatan VLAN, pengaturan IP per *interface* VLAN, pengaturan DHCP *Relay Agent*, dan penambahan OSPF *network* parameter untuk setiap VLAN. Terakhir, cuplikan hasil pembuatan konfigurasi di perangkat *switch* untuk lokasi Mataram, seperti terlihat pada gambar 11.

```
TASK [Variable untuk Switch di Setiap Lokasi] *****
ok: [localhost]

TASK [Task untuk Switch] *****
included: /etc/ansible/sw.yml for localhost
included: /etc/ansible/sw.yml for localhost
included: /etc/ansible/sw.yml for localhost

TASK [Pengaturan Mode VTP dan VTP Domain Name] *****
changed: [localhost -> 172.31.0.131] => (item={'vlanid': 2, 'name': 'MTH', 'portmembership': 'et0/0-3'})
changed: [localhost -> 172.31.0.131] => (item={'vlanid': 3, 'name': 'MTH', 'portmembership': 'et1/0-3'})
changed: [localhost -> 172.31.0.131] => (item={'vlanid': 4, 'name': 'MTH', 'portmembership': 'et2/0-3'})

TASK [Pengaturan Port Membership] *****
changed: [localhost -> 172.31.0.131] => (item={'vlanid': 2, 'name': 'MTH', 'portmembership': 'et0/0-3'})
changed: [localhost -> 172.31.0.131] => (item={'vlanid': 3, 'name': 'MTH', 'portmembership': 'et1/0-3'})
changed: [localhost -> 172.31.0.131] => (item={'vlanid': 4, 'name': 'MTH', 'portmembership': 'et2/0-3'})

TASK [Pengaturan Mode VTP dan VTP Domain Name] *****
changed: [localhost -> 172.31.0.132] => (item={'vlanid': 2, 'name': 'MTH', 'portmembership': 'et0/0-3'})
changed: [localhost -> 172.31.0.132] => (item={'vlanid': 3, 'name': 'MTH', 'portmembership': 'et1/0-3'})
changed: [localhost -> 172.31.0.132] => (item={'vlanid': 4, 'name': 'MTH', 'portmembership': 'et2/0-3'})

TASK [Pengaturan Port Membership] *****
changed: [localhost -> 172.31.0.132] => (item={'vlanid': 2, 'name': 'MTH', 'portmembership': 'et0/0-3'})
changed: [localhost -> 172.31.0.132] => (item={'vlanid': 3, 'name': 'MTH', 'portmembership': 'et1/0-3'})
changed: [localhost -> 172.31.0.132] => (item={'vlanid': 4, 'name': 'MTH', 'portmembership': 'et2/0-3'})

TASK [Pengaturan Mode VTP dan VTP Domain Name] *****
changed: [localhost -> 172.31.0.133] => (item={'vlanid': 2, 'name': 'MTH', 'portmembership': 'et0/0-3'})
changed: [localhost -> 172.31.0.133] => (item={'vlanid': 3, 'name': 'MTH', 'portmembership': 'et1/0-3'})
changed: [localhost -> 172.31.0.133] => (item={'vlanid': 4, 'name': 'MTH', 'portmembership': 'et2/0-3'})

TASK [Pengaturan Port Membership] *****
changed: [localhost -> 172.31.0.133] => (item={'vlanid': 2, 'name': 'MTH', 'portmembership': 'et0/0-3'})
changed: [localhost -> 172.31.0.133] => (item={'vlanid': 3, 'name': 'MTH', 'portmembership': 'et1/0-3'})
changed: [localhost -> 172.31.0.133] => (item={'vlanid': 4, 'name': 'MTH', 'portmembership': 'et2/0-3'})

PLAY RECAP *****
localhost : ok=32 changed=13 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

Gambar 11. Hasil Eksekusi Playbook untuk Task Membuat Konfigurasi di Switch Lokasi Mataram

Terlihat terdapat 4 (empat) *tasks* yang dieksekusi yaitu *variable* untuk *switch* di setiap lokasi, *task* untuk *switch*, pengaturan mode VTP dan VTP *domain name*, pengaturan *port membership*. Khusus untuk dua *task* terakhir yaitu pengaturan mode VTP dan VTP *domain name*, serta pengaturan *port membership* akan diulang sejumlah *switch* yang ada di lokasi Mataram yaitu 3 (tiga) *switch*. Gambar 12 memperlihatkan cuplikan hasil eksekusi *playbook* untuk mengotomatisasi penghapusan konfigurasi VLAN berbasis VTP dan DHCP di lokasi Mataram yang dipicu menggunakan perintah `ansible-playbook main_delete.yml -e lokasi=mataram`. Terlihat terdapat 5 (lima) *tasks* yang dieksekusi yaitu *include variable*,

variable untuk router di setiap lokasi, task untuk router, menghapus DHCP Server berupa dan menghapus DHCP Excluded Address.

```

root@ansible-workstation:~# ansible-playbook main_delete.yml -e lokasi=mataram
PLAY [router]
TASK [include variable]
ok: [localhost]
TASK [Variable untuk Router di Setiap Lokasi]
ok: [localhost]
TASK [Task untuk Router]
included: /etc/ansible/delete_config_router.yml for localhost
TASK [Menghapus DHCP Server]
changed: [localhost -> 172.31.0.1] => (item={'departemen': 'HRD_MTM', 'network': '172.31.0.32', 'mask': '255.255.255.224', 'default_router': '172.31.0.23'})
changed: [localhost -> 172.31.0.1] => (item={'departemen': 'MKT_MTM', 'network': '172.31.0.64', 'mask': '255.255.255.224', 'default_router': '172.31.0.65'})
changed: [localhost -> 172.31.0.1] => (item={'departemen': 'SALES_MTM', 'network': '172.31.0.96', 'mask': '255.255.255.224', 'default_router': '172.31.0.97'})
TASK [Menghapus DHCP Excluded Address]
changed: [localhost -> 172.31.0.1] => (item={'departemen': 'HRD_MTM', 'network': '172.31.0.32', 'mask': '255.255.255.224', 'default_router': '172.31.0.23'})
changed: [localhost -> 172.31.0.1] => (item={'departemen': 'MKT_MTM', 'network': '172.31.0.64', 'mask': '255.255.255.224', 'default_router': '172.31.0.65'})
changed: [localhost -> 172.31.0.1] => (item={'departemen': 'SALES_MTM', 'network': '172.31.0.96', 'mask': '255.255.255.224', 'default_router': '172.31.0.97'})
    
```

**Gambar 12. Hasil Eksekusi Playbook untuk Task Menghapus Konfigurasi Seluruh Perangkat di Lokasi Mataram**

Pengukuran waktu dilakukan dengan menggunakan *stopwatch* dari aplikasi *Alarm & Clock* yang terdapat pada *sistem operasi Windows* dimana diaktifkan ketika awal eksekusi suatu percobaan dan dihentikan ketika percobaan tersebut telah selesai dieksekusi. Gambar 13 memperlihatkan cuplikan salah satu hasil pengukuran waktu untuk percobaan pembuatan konfigurasi VLAN berbasis VTP dan DHCP di perangkat jaringan yang berlokasi di Mataram baik secara manual maupun otomatisasi.



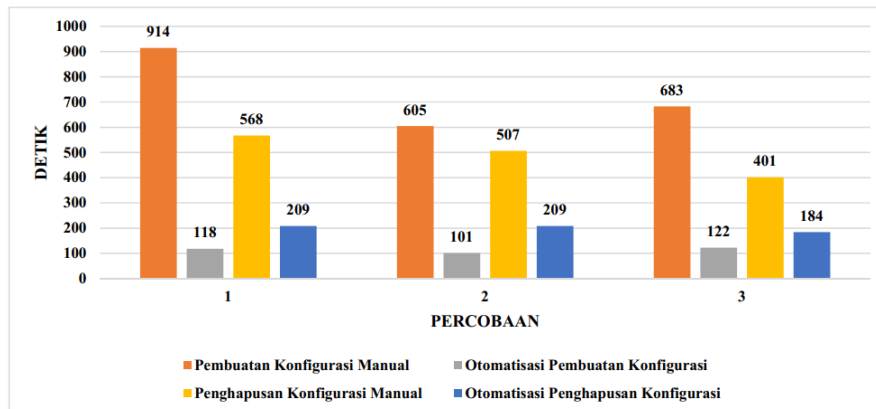
**Gambar 13. Pengukuran Waktu Eksekusi Pembuatan Konfigurasi VLAN berbasis VTP dan DHCP secara Manual dan Otomatisasi untuk Lokasi Mataram**

Terlihat pada hasil pengukuran waktu antara konfigurasi manual dan otomatisasi dalam pembuatan di lokasi Mataram. Pada percobaan pertama secara manual membutuhkan waktu 15 menit 14 detik, sedangkan secara otomatisasi membutuhkan waktu 1 menit 58 detik. Pada percobaan kedua secara manual membutuhkan waktu 10 menit 5 detik sedangkan secara otomatisasi membutuhkan waktu 1 menit 41 detik. Terakhir, pada percobaan ketiga secara manual membutuhkan waktu 11 menit 23 detik sedangkan secara otomatisasi membutuhkan waktu 2 menit 2 detik.

### 3.4. Analisa Hasil Ujicoba

Berdasarkan hasil ujicoba otomatisasi manajemen konfigurasi VLAN berbasis VTP dan DHCP yang telah dilakukan maka diperoleh analisa sebagai berikut:

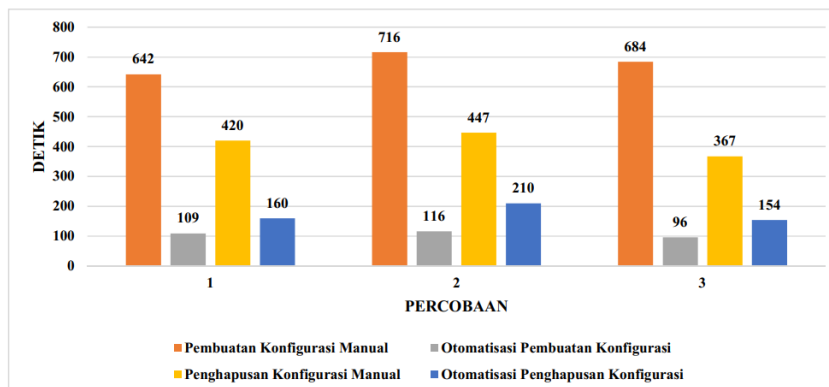
- a) Sistem otomatisasi *Ansible Playbook* yang dibangun dapat digunakan untuk manajemen konfigurasi perangkat jaringan baik *router*, *switch* maupun *multilayer switch* meliputi pembuatan dan penghapusan VLAN berbasis VTP dan DHCP di setiap lokasi yaitu Mataram, Denpasar dan Surabaya.
- b) Perbandingan waktu manajemen konfigurasi perangkat jaringan secara manual dan otomatisasi menggunakan *Ansible Playbook* di lokasi Mataram, seperti terlihat pada gambar 14.



**Gambar 14. Perbandingan Waktu Pembuatan dan Penghapusan Konfigurasi Secara Manual dan Otomatisasi di Lokasi Mataram**

Terlihat pembuatan konfigurasi secara manual di lokasi Mataram membutuhkan waktu terlama 914 detik atau 15 menit 14 detik pada percobaan pertama dan tercepat 605 detik atau 10 menit 5 detik pada percobaan kedua. Sebaliknya pada pembuatan konfigurasi menggunakan otomatisasi membutuhkan waktu terlama 122 detik atau 2 menit 2 detik pada percobaan ketiga dan tercepat 101 detik atau 1 menit 41 detik pada percobaan kedua. Perbandingan rata-rata waktu pembuatan konfigurasi sebanyak 3 (tiga) kali percobaan yang dilakukan secara manual yaitu 734 detik atau 12 menit 14 detik dan 114 detik atau 1 menit 54 detik secara otomatisasi. Selain itu juga memperlihatkan perbandingan rata-rata waktu penghapusan konfigurasi sebanyak 3 (tiga) kali percobaan yang dilakukan secara manual yaitu 492 detik atau 8 menit 12 detik dan 201 detik atau 3 menit 21 detik secara otomatisasi.

- c) Perbandingan waktu manajemen konfigurasi perangkat jaringan secara manual dan otomatisasi menggunakan *Ansible Playbook* di lokasi Denpasar, seperti terlihat pada gambar 15.

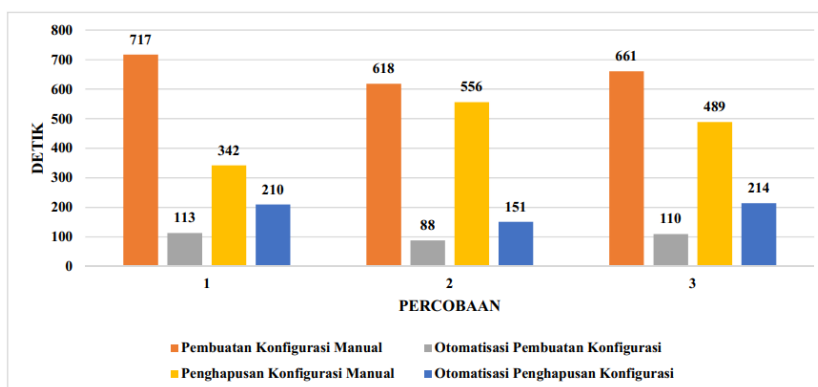


**Gambar 15. Perbandingan Waktu Pembuatan Konfigurasi Secara Manual dan Otomatisasi di Lokasi Denpasar**

Terlihat pembuatan konfigurasi secara manual di lokasi Denpasar membutuhkan waktu terlama 716 detik atau 11 menit 56 detik pada percobaan kedua dan tercepat 642 detik atau 10 menit 42 detik pada percobaan pertama. Sebaliknya pada pembuatan konfigurasi menggunakan otomatisasi membutuhkan waktu terlama 116 detik atau 1 menit 56 detik pada percobaan kedua dan tercepat 96 detik atau 1 menit 36 detik pada percobaan ketiga.

Perbandingan rata-rata waktu pembuatan konfigurasi sebanyak 3 (tiga) kali percobaan yang dilakukan secara manual yaitu 681 detik atau 11 menit 21 detik dan 107 detik atau 1 menit 47 detik secara otomatisasi. Selain itu juga memperlihatkan perbandingan rata-rata waktu penghapusan konfigurasi sebanyak 3 (tiga) kali percobaan yang dilakukan secara manual yaitu 412 detik atau 6 menit 52 detik dan 175 detik atau 2 menit 55 detik secara otomatisasi.

- d) Perbandingan waktu manajemen konfigurasi perangkat jaringan secara manual dan otomatisasi menggunakan *Ansible Playbook* di lokasi Surabaya, seperti terlihat pada gambar 16.



**Gambar 16. Perbandingan Waktu Pembuatan Konfigurasi Secara Manual dan Otomatisasi di Lokasi Surabaya**

Terlihat pembuatan konfigurasi secara manual di lokasi Surabaya membutuhkan waktu terlama 717 detik atau 11 menit 57 detik pada percobaan pertama dan tercepat 618 detik atau 10 menit 18 detik pada percobaan kedua. Sebaliknya pada pembuatan konfigurasi menggunakan otomatisasi membutuhkan waktu terlama 113 detik atau 1 menit 53 detik pada percobaan pertama dan tercepat 88 detik atau 1 menit 28 detik pada percobaan kedua. Perbandingan rata-rata waktu pembuatan konfigurasi sebanyak 3 (tiga) kali percobaan yang dilakukan secara manual yaitu 666 detik atau 11 menit 6 detik dan 104 detik atau 1 menit 44 detik secara otomatisasi. Selain itu juga memperlihatkan perbandingan rata-rata waktu penghapusan konfigurasi sebanyak 3 (tiga) kali percobaan yang dilakukan secara manual yaitu 463 detik atau 7 menit 42 detik dan 192 detik atau 3 menit 12 detik secara otomatisasi.

- e) Pembuatan konfigurasi VLAN berbasis VTP dan layanan DHCP secara manual memiliki waktu rata-rata 693.3 detik atau 11 menit 33 detik dan secara otomatisasi yaitu 108.1 detik atau 1 menit 48 detik berdasarkan 3 (tiga) kali percobaan di setiap lokasi. Sedangkan penghapusan konfigurasi secara manual memiliki waktu rata-rata 455.2 detik atau 7 menit 35 detik sedangkan secara otomatisasi yaitu 189 detik atau 3 menit 9 detik.
- f) Faktor yang mempengaruhi pembuatan dan penghapusan konfigurasi secara manual lebih lama daripada otomatisasi adalah banyaknya perintah konfigurasi VLAN berbasis VTP dan layanan DHCP yang harus dimasukkan pada setiap perangkat baik *router*, *multilayer switch* dan *switch* di setiap lokasi. Waktu yang diperlukan berbanding lurus dengan jumlah perintah dari kebijakan konfigurasi yang diterapkan. Selain itu juga dipengaruhi oleh kecepatan pengetikan konfigurasi dan kesalahan pengetikan perintah atau parameter ketika melakukan konfigurasi. Sebaliknya pada sistem otomatisasi, penyesuaian hanya dilakukan pada *variable files* yang menyimpan nilai dari kebijakan konfigurasi secara dinamis dari setiap lokasi tanpa perintah konfigurasi sehingga dapat mempercepat proses pembuatan dan penghapusan konfigurasi.

**4. Kesimpulan**

Adapun kesimpulan yang dapat diambil berdasarkan hasil ujicoba yang telah dilakukan yaitu *Ansible playbook* yang dibuat berhasil digunakan untuk melakukan otomatisasi secara dinamis dalam pembuatan dan penghapusan VLAN berbasis VTP dan layanan DHCP menggunakan modul *ios\_config*. Selain itu sistem otomatisasi yang dibuat dapat mempercepat

proses pengelolaan VLAN berdasarkan 3 (tiga) kali percobaan di setiap lokasi yaitu rata-rata 6 (enam) kali lebih cepat ketika operasi pembuatan konfigurasi yaitu lebih cepat 585.3 detik atau 9 menit 45 detik dibandingkan dengan cara manual. Sedangkan ketika operasi penghapusan konfigurasi, sistem otomatisasi rata-rata 2 (dua) kali lebih cepat dibandingkan secara manual yaitu 266.3 detik atau 4 menit 26 detik dibandingkan dengan cara manual.

Adapun saran-saran untuk pengembangan penelitian ini lebih lanjut adalah mengembangkan sistem otomatisasi manajemen *backup* konfigurasi dan pembaharuan sistem operasi dari perangkat jaringan baik *router*, *switch* maupun *multilayer switch*.

### Ucapan Terima Kasih

Terimakasih kepada Universitas Bumigora atas dukungan sarana dan prasarana dalam penyelesaian penelitian ini. Selain itu disampaikan terimakasih pula kepada tim jurnal BITE yang telah memberikan kesempatan untuk mempublikasikan hasil penelitian yang telah dilakukan.

### Referensi

- [1] O. K. Sulaiman, "Simulasi Perancangan Sistem Jaringan Inter Vlan Routing di Universitas Negeri Medan," *CESS (Journal Comput. Eng. Syst. Sci.*, vol. 2, no. 3, pp. 92–96, 2017.
- [2] D. A. J. Al-khaffaf, "Improving LAN Performance Based on IEEE802 . 1Q VLAN Switching Techniques," *J. Univ. Babylon*, no. 1, pp. 286–297, 2018.
- [3] S. Somasundaram and M. Chandran, "A Simulation based study on Network Architecture Using Inter-VLAN Routing and Secure Campus Area Network (CAN)," *Int. J. Comput. Sci. Eng.*, vol. 6, no. 3, pp. 111–121, 2018.
- [4] P. H. Sutanto, "Analisis Perancangan Virtual Local Area Network Berbasis Vtp Dan Inter-Vlan Routing Pada Perusahaan Daerah Air Minum Tirta," *J. Tek. Komput.*, vol. IV, no. 2, pp. 125–134, 2018.
- [5] M. Mubiatna, R. Azhar, and K. Marzuki, "Otomatisasi Manajemen Konfigurasi Vlan Intervlan dan Dhep Server pada Cisco menggunakan Ansible," 2018.
- [6] P. Masek, M. Stusek, J. Krejci, K. Zeman, J. Pokorny, and M. Kudlacek, "Unleashing full potential of ansible framework: University labs administration," *Conf. Open Innov. Assoc. Fruct*, vol. 2018-May, pp. 144–150, 2018.
- [7] M. A. A. Pratama and I. P. Hariyadi, "Otomasi Manajemen dan Pengawasan Linux Container (LCX) Pada Proxmox VE Menggunakan Ansible," *J. Bumigora Inf. Technol.*, vol. 3, no. 1, pp. 82–95, 2021.
- [8] R. David Marcus, H. Eka Rosyadi, and F. Yulian Pamuji, "Prototype Sistem Administrasi Dan Keamanan Jaringan Komputer Berbasis DHCP Server Mikrotik," vol. 6, no. 62, pp. 685–695, 2021.
- [9] I. P. Hariyadi and A. Juliansyah, "Analisa Penerapan Private Cloud Computing Berbasis Proxmox Virtual Environment Sebagai Media Pembelajaran Praktikum Manajemen Jaringan," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 18, no. 1, pp. 1–12, 2018.
- [10] D. Siswanto, G. Priyandoko, N. Tjahjono, R. S. Putri, N. B. Sabela, and M. I. Muzakki, "Development of Information and Communication Technology Infrastructure in School using an Approach of the Network Development Life Cycle Method," *J. Phys. Conf. Ser.*, vol. 1908, no. 1, 2021.

