

Otomatisasi Konfigurasi VLAN pada Open vSwitch Menggunakan Python Scripting dalam Lingkungan Simulasi GNS3

Febrianta Immanuel Purba¹ Deddy Kiswanto² M Farel Revansha Buulolo³ Hafizham Mufti Siregar⁴

Program Studi Ilmu Komputer, Universitas Negeri Medan, Kota Medan, Provinsi Sumatera Utara, Indonesia^{1,2,3}

Email: febriantapurba445@gmail.com¹ dedykiswanto@unimed.ac.id²
farelrevansha@gmail.com³ hafizhammufti123@gmail.com⁴

Abstrak

Pengelolaan konfigurasi jaringan secara manual melalui Command Line Interface (CLI) merupakan pendekatan yang memakan waktu dan rentan terhadap kesalahan manusia (human error), khususnya pada proses konfigurasi Virtual Local Area Network (VLAN) di lingkungan jaringan virtual. Penelitian ini bertujuan merancang dan mengimplementasikan sistem otomatisasi konfigurasi VLAN pada Open vSwitch (OVS) menggunakan Python scripting dalam lingkungan simulasi GNS3. Metode pengembangan yang digunakan adalah Network Development Life Cycle (NDLC) yang mencakup tahapan Analysis, Design, Simulation Prototype, Implementation, Monitoring, dan Management. Topologi jaringan yang dibangun terdiri dari satu perangkat switch bawaan GNS3, dua perangkat Open vSwitch, enam host virtual menggunakan Virtual PC Simulator (VPCS), dan satu node Cloud sebagai penghubung ke komputer administrator. Python script dieksekusi dari komputer administrator menggunakan Visual Studio Code melalui koneksi SSH berbasis library Netmiko, mengonfigurasi bridge br0 serta mendaftarkan port eth1 ke VLAN 10, eth2 ke VLAN 20, dan eth3 ke VLAN 30 melalui perintah ovs-vsctl. Pengujian dilakukan menggunakan metode Black Box Testing dengan 10 skenario fungsional. Hasil pengujian menunjukkan seluruh skenario berhasil dengan status lulus, meliputi pembuatan bridge, pendaftaran port, verifikasi konfigurasi, komunikasi intra-VLAN, dan isolasi antar-VLAN. Pendekatan Python scripting berbasis SSH terbukti efektif, terstruktur, dan mampu mengotomatiskan konfigurasi jaringan virtual tanpa memerlukan instalasi agen tambahan di sisi perangkat OVS.

Kata Kunci: VLAN, Open vSwitch, Skrip Python, GNS3, Otomatisasi Jaringan

Abstract

Manual network configuration management through the Command Line Interface (CLI) is a time consuming approach and prone to human error, particularly when configuring Virtual Local Area Networks (VLANs) in a virtual network environment. This study aims to design and implement a VLAN configuration automation system on Open vSwitch (OVS) using Python scripting in a GNS3 simulation environment. The development method used is the Network Development Life Cycle (NDLC), which includes the stages of Analysis, Design, Simulation Prototype, Implementation, Monitoring, and Management. The network topology constructed consists of one GNS3-enabled switch, two Open vSwitch devices, six virtual hosts using Virtual PC Simulator (VPCS), and one Cloud node as a connection to the administrator's computer. The Python script was executed from the administrator's computer using Visual Studio Code over an SSH connection based on the Netmiko library. The script configured the br0 bridge and registered port eth1 to VLAN 10, eth2 to VLAN 20, and eth3 to VLAN 30 using the ovs-vsctl command. Testing was conducted using the Black Box Testing method with 10 functional scenarios. Test results showed all scenarios were successfully executed with a pass status, including bridge creation, port registration, configuration verification, intra-VLAN communication, and inter-VLAN isolation. The SSH-based Python scripting approach proved effective, structured, and capable of automating virtual network configuration without requiring additional agent installation on the OVS device side.

Keywords: VLAN, Open vSwitch, Python Scripting, GNS3, Network Automation



This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/).

PENDAHULUAN

Revolusi industri 4.0 ditandai dengan semakin besarnya pengaruh teknologi dalam kehidupan manusia. Era ini mendisrupsi berbagai bidang aktivitas manusia di antaranya bidang politik, sosial, ekonomi, pendidikan, kesehatan, dan masih banyak lainnya. Besarnya dampak yang dirasakan dari revolusi industri 4.0 saat ini sangat sulit untuk ditolak bahkan oleh seluruh umat manusia di dunia, khususnya di Indonesia (Kiswanto, D., Syahputra, H., & Panggabean, 2023). Perkembangan teknologi informasi dan komunikasi yang pesat telah mendorong meningkatnya kompleksitas infrastruktur jaringan komputer di berbagai organisasi. Pengelolaan jaringan yang terdiri dari banyak perangkat aktif menjadi tantangan tersendiri bagi administrator jaringan. Konfigurasi yang dilakukan secara manual melalui antarmuka Command Line Interface (CLI) tidak hanya memakan waktu yang sangat lama, tetapi juga rentan terhadap kesalahan manusia (human error) yang dapat berdampak pada gangguan layanan jaringan secara keseluruhan. Rahardika dan Ratama (2021) menunjukkan bahwa konfigurasi perangkat jaringan secara konvensional yang dilakukan satu per satu membutuhkan waktu dan usaha yang besar, sekaligus meningkatkan risiko kekeliruan yang umumnya dilakukan oleh manusia. Firmansyah (2024) memperkuat temuan tersebut dengan melaporkan bahwa pada infrastruktur jaringan yang kompleks, proses konfigurasi manual dapat memakan waktu hingga tiga hari kerja dan secara signifikan meningkatkan potensi human error pada setiap tahapnya. Seiring meningkatnya kompleksitas jaringan modern, kebutuhan terhadap tenaga profesional yang memiliki kompetensi di bidang jaringan juga semakin tinggi. Solihin et al. (Achmad Fauzi, Desi, 2026) menegaskan bahwa kompleksitas jaringan merupakan tantangan utama bagi administrator dalam mengelola banyak perangkat, melakukan pemeliharaan, dan menghadapi ancaman keamanan yang semakin canggih, sehingga mendorong adopsi pendekatan yang lebih efisien dalam pengelolaan infrastruktur jaringan.

Virtual Local Area Network (VLAN) adalah salah satu teknologi dalam manajemen jaringan yang memungkinkan segmentasi jaringan tanpa mengubah infrastruktur fisik sehingga memberikan peningkatan keamanan, bandwidth yang lebih efisien dan mempermudah pengelolaan. Sedangkan Open vSwitch (OVS) merupakan implementasi switch virtual multilayer yang bersifat open-source dan berlisensi Apache 2.0. OVS dirancang untuk mendukung otomatisasi jaringan melalui ekstensi terprogram (programmatic extension), sambil tetap mendukung antarmuka manajemen standar seperti OpenFlow dan protokol IEEE 802.1Q. Penelitian terkait network automation telah banyak dilakukan dalam beberapa tahun terakhir. Mauboy dan Wellem (2022) membuktikan bahwa implementasi network automation menggunakan library Python seperti Paramiko dan Netmiko mampu mengotomatiskan berbagai skenario konfigurasi perangkat jaringan secara efisien dalam lingkungan simulasi GNS3. Alfaresa et al. (2023) lebih lanjut memperlihatkan bahwa library otomatisasi jaringan berbasis Python dapat digunakan untuk mengonfigurasi protokol routing IGP maupun EGP secara otomatis, menggantikan proses manual yang berulang. Ardiansyah et al. (2025) juga mengonfirmasi dalam penelitiannya bahwa library Netmiko pada python dapat digunakan untuk mempermudah pemantauan dan pemeliharaan perangkat jaringan. Selain itu, Manik et al. (2023) membuktikan bahwa GNS3 mampu merancang dan mensimulasikan topologi jaringan menggunakan teknologi virtualisasi sebelum implementasi nyata, sehingga menjadi lingkungan yang sesuai untuk pengujian otomatisasi jaringan.

Berdasarkan penelitian sebelumnya, implementasi network automation telah banyak dilakukan menggunakan Ansible maupun Netmiko pada berbagai perangkat jaringan dan lingkungan simulasi GNS3. Namun, penelitian yang secara khusus membahas otomatisasi konfigurasi VLAN pada Open vSwitch menggunakan Python scripting dalam lingkungan

simulasi GNS3 masih terbatas. Penelitian Marzuki dkk. (2023) telah mengimplementasikan otomatisasi konfigurasi Open vSwitch menggunakan Ansible pada Proxmox, sedangkan Firmansyah (2024) dan Syah dkk. (2020) hanya fokus pada network automation di GNS3 dan tidak membahas konfigurasi VLAN pada Open vSwitch. Selain itu, sebagian besar isi penelitian yang telah dilakukan menitikberatkan pada otomatisasi jaringan secara umum, sehingga belum secara spesifik membahas implementasi otomatisasi konfigurasi VLAN pada Open vSwitch menggunakan Python scripting yang dijalankan dari komputer host/administrator melalui koneksi SSH. Oleh karena itu, penelitian ini berperan mengisi kesenjangan tersebut dengan mengembangkan mekanisme otomatisasi konfigurasi VLAN berbasis Python scripting pada Open vSwitch yang berjalan di lingkungan simulasi GNS3. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem otomatisasi konfigurasi VLAN pada Open vSwitch menggunakan Python scripting dengan library Netmiko yang dieksekusi dari komputer host/administrator dalam lingkungan simulator GNS3. Diharapkan penelitian ini dapat memberikan solusi untuk menyederhanakan pengelolaan jaringan virtual yang semakin kompleks, sekaligus menjadi rujukan implementasi network automation berbasis open-source tool yang ringan dan mudah diterapkan.

Penelitian Terdahulu yang Relevan

No	Penulis/ Tahun	Judul	Tujuan	Metode	Hasil	Kontribusi dan relevansi
1	Marzuki, K., dkk. (2023)	Automation of Open VSwitch-Based Virtual Network Configuration Using Ansible on Proxmox Virtual Environment	Mengotomatisasi konfigurasi jaringan virtual berbasis Open vSwitch pada lingkungan Proxmox.	NDLC dan otomatisasi menggunakan Ansible.	Otomatisasi berhasil mempercepat konfigurasi jaringan virtual dan VLAN pada Open vSwitch.	sama-sama menggunakan Open vSwitch. Perbedaannya penelitian ini menggunakan Ansible pada Proxmox,
2	Iqbal Firmansyah (2024)	Implementasi Network Automation Pada GNS3 Untuk Otomasi Konfigurasi Di PT. Semut Merah Squad	Menerapkan network automation untuk mengurangi konfigurasi manual perangkat jaringan.	Metode kualitatif dengan prototipe network automation berbasis Python pada GNS3.	Network automation mampu meningkatkan efisiensi konfigurasi perangkat jaringan.	menggunakan Python dan GNS3. Namun penelitian ini tidak membahas Open vSwitch maupun VLAN Automation secara khusus.
3	Syah, I., dkk. (2020)	Simulasi Network Automation Menggunakan Ansible Di GNS3 (Studi Kasus Smile Project)	Mengimplementasikan otomatisasi konfigurasi jaringan pada lingkungan simulasi GNS3.	Simulasi network automation menggunakan Ansible.	Otomatisasi mampu mengurangi waktu konfigurasi dan meminimalkan kesalahan konfigurasi.	menggunakan GNS3 untuk simulasi otomatisasi jaringan. Perbedaannya penelitian ini menggunakan Ansible dan tidak fokus pada VLAN maupun Open vSwitch.
4	Yalestia Chandrawaty & Hariyadi (2021)	Implementasi Ansible Playbook Untuk Mengotomatisasi Manajemen Konfigurasi VLAN Berbasis VTP dan Layanan DHCP	Mengotomatisasi pengelolaan VLAN dan layanan DHCP pada jaringan Cisco.	Implementasi Ansible Playbook.	VLAN dan DHCP dapat dikonfigurasi secara otomatis dan konsisten.	membahas otomatisasi VLAN. Perbedaannya menggunakan perangkat Cisco dan Ansible.

5	Mauboy & Wellem (2022)	Studi Perbandingan Library Untuk Implementasi Network Automation Menggunakan Paramiko dan Netmiko Pada Router Mikrotik	Membandingkan performa library Paramiko dan Netmiko untuk otomatisasi jaringan.	Eksperimen dan pengujian performa library.	Netmiko lebih efisien dan mudah digunakan untuk otomatisasi konfigurasi perangkat jaringan.	menjadi dasar pemilihan library otomatisasi jaringan berbasis Python yang dapat diterapkan pada penelitian ini.
---	------------------------	--	---	--	---	---

METODE PENELITIAN

Penelitian ini menggunakan metode penelitian kualitatif dengan pendekatan studi kasus dan deskriptif untuk mengkaji penerapan otomatisasi konfigurasi VLAN pada Open vSwitch (OVS) dalam lingkungan simulasi GNS3. Data penelitian diperoleh melalui observasi terhadap proses konfigurasi jaringan serta dokumentasi hasil implementasi dan pengujian sistem. Metodologi pengembangan jaringan yang digunakan adalah Network Development Life Cycle (NDLC), yang terdiri dari tahapan Analysis, Design, Simulation Prototype, Implementation, Monitoring, dan Management. Metode ini dipilih karena sesuai untuk perancangan, pembangunan, serta evaluasi sistem jaringan yang melibatkan otomatisasi konfigurasi perangkat jaringan berbasis Open vSwitch (OVS).

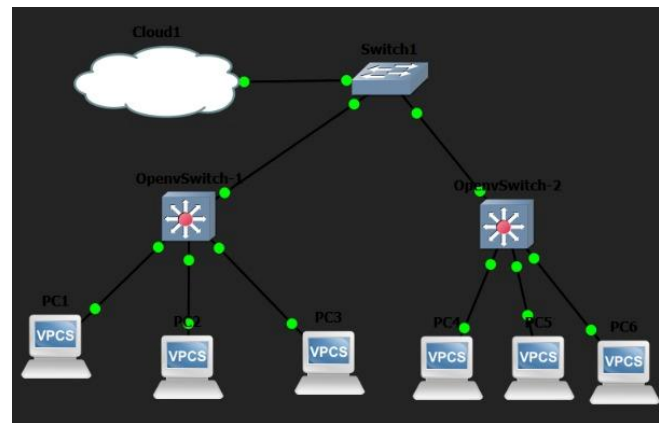
Analysis

Pada tahap analisis dilakukan identifikasi kebutuhan sistem dan permasalahan yang akan diselesaikan. Permasalahan utama yang ditemukan adalah proses konfigurasi VLAN pada perangkat jaringan masih dilakukan secara manual melalui Command Line Interface (CLI), sehingga membutuhkan waktu lebih lama dan berpotensi menimbulkan kesalahan konfigurasi. Kebutuhan sistem yang diperlukan meliputi:

- Perangkat Open vSwitch (OVS) Berjalan sebagai kontainer *lightweight* minimalis di dalam lingkungan emulator GNS3 sebagai core switch yang akan dikonfigurasi.
- Perangkat switch bawaan GNS3
- Virtual PC Simulator (VPCS) sebagai host uji pada masing-masing VLAN.
- Komputer administrator (Windows) Terhubung ke dalam topologi jaringan GNS3 melalui perangkat perantara node Cloud (memanfaatkan *VirtualBox Host-Only Network Adapter*) untuk kebutuhan manajemen jalur lokal.
- Text Editor: Digunakan oleh administrator untuk menulis, menyunting, dan menyimpan kode instruksi di dalam repositori lokal Open vSwitch.
- Python script: Sebagai instrumen utama otomatisasi yang dieksekusi dari komputer administrator menggunakan Visual Studio Code, mengirimkan serangkaian perintah `ovs-vsctl` ke perangkat Open vSwitch melalui koneksi SSH berbasis library Netmiko

Design

Tahap desain dilakukan dengan merancang topologi jaringan yang digunakan dalam simulasi menggunakan GNS3. Topologi terdiri satu perangkat switch bawaan dari GNS3, dua perangkat Open vSwitch yang terhubung dengan masing masing 3 host virtual (PC1, PC2, PC3) dan (PC4, PC5, PC6) serta satu Cloud (Host-Only) untuk konektivitas eksternal menuju komputer administrator. Seluruh host (PC1–PC6) dikonfigurasi dalam satu subnet yang sama, yaitu 192.168.10.X/24, dengan pertimbangan bahwa pengujian dalam penelitian ini difokuskan pada validasi isolasi layer 2 berbasis VLAN tag (IEEE 802.1Q), bukan pada segmentasi layer 3. Dalam konteks ini, isolasi antar-VLAN tetap berfungsi karena Open vSwitch memisahkan domain broadcast, sehingga host lintas VLAN tidak dapat saling berkomunikasi meskipun berada dalam rentang alamat IP yang sama.



Gambar 1. Topologi GNS3

Rancangan Topologi Jaringan:

1. 1 Perangkat switch bawaan sebagai simpul switching utama dalam jaringan
2. 2 Perangkat Open vSwitch
3. 6 Perangkat host virtual (PC1, PC2, PC3, PC4, PC5, dan PC6) Menggunakan *Virtual PC Simulator* (VPCS) yang dihubungkan ke port fisik OVS (eth1, eth2, eth3, dan eth4)
pc 1 : 192.168.10.1
pc 2 : 192.168.10.2
pc 3 : 192.168.10.3
pc 4 : 192.168.10.4
pc 5 : 192.168.10.5
pc 6 : 192.168.10.6
4. 1 Perangkat Cloud (Host-Only) terhubung ke port eth0 pada switch sebagai penghubung komunikasi windows dengan GNS3.

Simulation Prototype

Setelah desain selesai, dilakukan simulasi menggunakan GNS3 untuk memastikan rancangan dapat berjalan sesuai kebutuhan. Aktivitas pada tahap ini meliputi:

1. Pembuatan topologi jaringan pada GNS3 yang terdiri dari Switch bawaan, OpenvSwitch-1 dan 2, Cloud, PC1, PC2, PC3, PC4, PC5, dan PC6.
2. Saat menambahkan node CLOUD ke lembar kerja di GNS3 muncul pilihan server, maka pilih server local "Run on my local computer" bukan GNS3VM.
3. Konfigurasi IP Manajemen OVS via DHCP, Hubungkan kabel dari cloud ke OVS pada port eth0 lalu minta ip otomatis yang satu segmen dengan komputer melalui perintah "udhcpc -I eth0"
4. Instalasi paket OpenSSH (apk add openssh) pada sistem operasi berbasis Alpine milik OVS, melakukan generate host keys, mengonfigurasi hak akses root pada sshd_config, serta mengaktifkan daemon SSH melalui perintah /usr/sbin/sshd agar OVS siap menerima koneksi remote.
5. Pengujian ping dari komputer administrator Windows menuju ip OVS 1 dan 2 (192.168.XX.XX) untuk memastikan interkoneksi local terjalin sempurna.

Tahap simulasi bertujuan untuk mengurangi risiko kesalahan sebelum implementasi dilakukan secara penuh.

Implementation

Tahap implementasi dilakukan dengan mengembangkan script otomatisasi konfigurasi VLAN langsung dengan Python Scripting melalui pada Text editor computer seperti VS Code

```
1 from netmiko import ConnectHandler
2 import getpass
3
4 print("="*50)
5 print("      RUNNING: AUTOMATION VLAN OPEN VSWITCH")
6 print("="*50)
7
8 username = "root"
9 password = getpass.getpass("Masukkan Password Root OVS: ")
10
11 # IP asli sesuai hasil running CMD kamu sebelumnya
12 daftar_ovs = [
13     '192.168.146.131', # OVS-1
14     '192.168.146.132', # OVS-2
15 ]
16 # Perintah OVS untuk memberikan tag VLAN langsung ke port yang berada di br0
17 perintah_ovs = [
18     'ovs-vsctl set port eth1 tag=10',
19     'ovs-vsctl set port eth2 tag=20',
20     'ovs-vsctl set port eth3 tag=30',
21     'ovs-vsctl show'
22 ]
23 for ip in daftar_ovs:
24     print("\n" + "-"*50)
25     print(f"--> Menghubungkan ke Open vSwitch: {ip}...")
26
27     device_profile = {
28         'device_type': 'linux',
29         'host': ip,
30         'username': username,
31         'password': password,
32     }
33     try:
34         connection = ConnectHandler(**device_profile)
35         print("      [Sukses] Berhasil login via SSH!")
36         print("      [Proses] Mengirimkan perintah konfigurasi VLAN...")
37
38         for cmd in perintah_ovs:
39             output = connection.send_command(cmd)
40             # Kita cetak output spesifik saat perintah 'ovs-vsctl show' dieksekusi
41             if 'show' in cmd:
42                 print(f"\n[Hasil Akhir OVS {ip}]:\n{output}")
43
44             connection.disconnect()
45             print(f"      [Selesai] Switch {ip} berhasil dikonfigurasi!")
46
47     except Exception as e:
48         print(f"      [Gagal] Tidak bisa mengonfigurasi {ip}. Error: {e}")
49
50 print("\n" + "-"*50)
51 print("PROSES SELESAI: Semua port telah dikonfigurasi VLAN!")
52 print("="*50)
```

Gambar 2. Kode Program Python

Python script dikembangkan dan dieksekusi dari komputer administrator menggunakan Visual Studio Code. Script membangun koneksi SSH ke masing-masing perangkat Open vSwitch menggunakan library Netmiko, kemudian mengirimkan perintah ovs-vsctl secara otomatis untuk melakukan konfigurasi VLAN. Konfigurasi yang diimplementasikan meliputi:

- Pendaftaran port eth1 ke bridge br0 dengan tag VLAN 10.
- Pendaftaran port eth2 ke bridge br0 dengan tag VLAN 20.
- Pendaftaran port eth3 ke bridge br0 dengan tag VLAN 30.

Monitoring

Tahap monitoring dilakukan untuk memastikan konfigurasi VLAN berhasil diterapkan pada perangkat Open vSwitch. Proses monitoring dilakukan dengan:

- Menjalankan kode program, perintah ovs-vsctl show pada OVS untuk melihat konfigurasi bridge, port, dan tag VLAN yang aktif.
- Memverifikasi keberadaan tag VLAN pada masing-masing port (eth1=10, eth2=20, eth3=30).

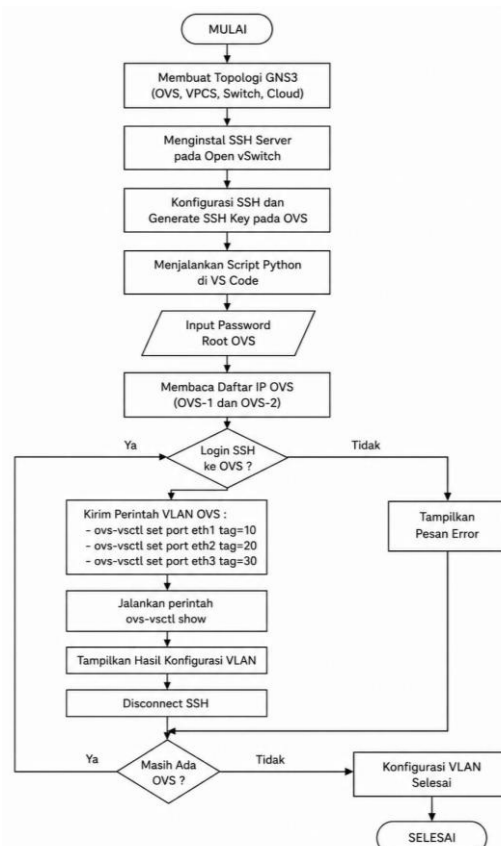
- Menguji komunikasi antar host dalam VLAN yang sama eth1 PC1 dan PC4 juga eth2 PC2 dan PC5 dan eth3 PC3 dan PC6 menggunakan perintah ping di VPCS (berhasil: 84 bytes received).

Management

Tahap management dilakukan untuk memastikan sistem dapat digunakan dan dikembangkan secara berkelanjutan. Aktivitas yang dilakukan meliputi:

- Dokumentasi topologi jaringan GNS3 beserta konfigurasi perangkat.
- Penyimpanan source code python script untuk keperluan replikasi dan pengembangan.
- Pencatatan hasil pengujian Black Box Testing sebagai bukti validasi sistem. .
- Pengembangan script untuk mendukung otomatisasi konfigurasi VLAN tambahan maupun fitur jaringan lainnya pada Open vSwitch.

Alur Kerja Sistem



Gambar 3. Flowchart Kerja Sistem

Pada gambar 3 terdapat Flowchart alur kerja otomatisasi konfigurasi VLAN pada Open vSwitch menggunakan Python script. Proses dimulai dengan pembuatan topologi jaringan pada GNS3, dilanjutkan instalasi dan konfigurasi Open vSwitch di GNS3 agar bisa terkoneksi ke komputer host/administrator. Selanjutnya, menjalankan Python script di Virtual Studio Code di komputer host/administrator yang kemudian menjalin koneksi SSH ke Open vSwitch dengan menggunakan IP yang telah ditentukan. Jika proses login berhasil, script akan mengirimkan serangkaian perintah `ovs-vsctl` yang kemudian otomatis mengkonfigurasi VLAN, kemudian menampilkan hasil konfigurasi yang telah diterapkan. Setelah proses selesai, koneksi SSH akan diputus dan setelah seluruh perangkat telah terkonfigurasi, maka proses otomatisasi VLAN dinyatakan selesai.

HASIL PENELITIAN DAN PEMBAHASAN

Hasil penelitian

Hasil Implementasi Python Script

Implementasi otomatisasi konfigurasi VLAN dilakukan melalui Python script yang dikembangkan menggunakan Visual Studio Code di komputer administrator. Script dieksekusi dari komputer administrator dan mengirimkan perintah ovs-vsctl ke perangkat Open vSwitch melalui koneksi SSH berbasis library Netmiko. Hasil yang diimplementasikan adalah sebagai berikut:

Tabel 1. Hasil Eksekusi Script Python

Parameter	Hasil
Koneksi SSH ke OVS	Berhasil
Pengiriman perintah ovs-vsctl	Berhasil
Pembuatan bridge OVS	Berhasil
Konfigurasi VLAN 10	Berhasil
Konfigurasi VLAN 20	Berhasil
Konfigurasi VLAN 30	Berhasil
Pesan error saat eksekusi	Tidak ada
Status implementasi	Berhasil

Pada tabel 1 diatas, dinyatakan bahwa setiap langkah otomatisasi konfigurasi VLAN di Open vSwitch telah berhasil dilakukan tanpa ada error. Netmiko berjalan dengan baik dalam membangun koneksi SHH sehingga perintah otomatisasi konfigurasi VLAN dapat terkirim dan dieksekusi oleh OVS . Hasil ini menunjukkan konfigurasi bridge dan VLAN yang telah ditentukan berhasil diterapkan sesuai rancangan.

Hasil Verifikasi Konfigurasi VLAN

Setelah script dieksekusi, verifikasi dilakukan menggunakan perintah ovs-vsctl show. Output menunjukkan bridge br0 berhasil dibuat dengan konfigurasi port dan tag VLAN sebagai berikut:

Tabel 2. Konfigurasi Port dan Tag VLAN

Port	VLAN	IP Host
eth1	10	192.168.10.1 dan 192.168.10.4
eth2	20	192.168.10.2 dan 192.168.10.5
eth3	30	192.168.10.3 dan 192.168.10.6

Seluruh port terdaftar pada bridge br0 dengan tag VLAN yang sesuai dengan rancangan pada tahap design. Hasil ini membuktikan bahwa satu eksekusi script mampu mengonfigurasi tiga port sekaligus secara otomatis dan konsisten.

Hasil Pengujian Black Box Testing

Pengujian fungsional dilakukan menggunakan metode Black Box Testing untuk memverifikasi apakah setiap fungsi utama sistem menghasilkan output yang sesuai dengan spesifikasi yang diharapkan. Seluruh 10 skenario pengujian menghasilkan output yang sesuai dengan spesifikasi yang diharapkan, sehingga sistem dinyatakan lulus uji fungsional secara keseluruhan bisa dilihat pada tabel berikut:

Tabel 3. Hasil Black Box Testing

No	Fungsi yang Diuji	Input	Output yang Diharapkan	Output Aktual	Status
1	Pendaftaran port VLAN 10	ovs-vsctl set port eth1 tag=10	eth1 terdaftar dengan tag VLAN 10	eth1 terdaftar dengan tag 10	Berhasil

2	Pendaftaran port VLAN 20	ovs-vsctl set port eth2 tag=20	eth2 terdaftar dengan tag VLAN 20	eth2 terdaftar dengan tag 20	Berhasil
3	Pendaftaran port VLAN 30	ovs-vsctl set port eth3 tag=30	eth3 terdaftar dengan tag VLAN 30	eth3 terdaftar dengan tag 30	Berhasil
4	Uji Ping VLAN 10	Ping PC1 → PC4	Paket diterima dengan sukses	84 bytes received	Berhasil
5	Uji Ping VLAN 20	Ping PC2 → PC5	Paket diterima dengan sukses	84 bytes received	Berhasil
6	Uji Ping VLAN 30	Ping PC3 → PC6	Paket diterima dengan sukses	84 bytes received	Berhasil
7	Uji Ping VLAN 10 dan 20	Ping PC1 → PC2	Paket gagal diterima	Host not reachable	Berhasil
8	Uji Ping VLAN 10 dan 30	Ping PC1 → PC3	Paket gagal diterima	Host not reachable	Berhasil
9	Uji Ping VLAN 10 dan 20	Ping PC1 → PC5	Paket gagal diterima	Host not reachable	Berhasil
10	Uji Ping VLAN 10 dan 30	Ping PC1 → PC6	Paket gagal diterima	Host not reachable	Berhasil

Pembahasan

Hasil implementasi menunjukkan bahwa Python scripting yang dieksekusi dari komputer administrator melalui koneksi SSH menggunakan library Netmiko mampu mengotomatiskan konfigurasi bridge, port, dan tag VLAN pada Open vSwitch secara penuh hanya dengan satu kali eksekusi script. Seluruh perintah ovs-vsctl berhasil dijalankan tanpa error dan konfigurasi yang dihasilkan sesuai dengan rancangan sistem. Temuan ini menjawab permasalahan yang diangkat pada penelitian mengenai kompleksitas dan tingginya risiko kesalahan pada konfigurasi jaringan secara manual. Hasil tersebut sejalan dengan penelitian Mazin et al. (2021) yang menyatakan bahwa otomatisasi jaringan berbasis Python mampu mengurangi ketergantungan terhadap proses konfigurasi manual yang memakan waktu dan berpotensi menimbulkan human error. Keberhasilan otomatisasi ini didukung oleh penggunaan library Netmiko yang mampu membangun koneksi SSH secara langsung ke perangkat Open vSwitch dan mengeksekusi perintah konfigurasi secara berurutan. Hasil pengujian komunikasi antar host menunjukkan bahwa segmentasi VLAN berjalan sesuai rancangan, di mana host yang berada pada VLAN yang sama dapat saling berkomunikasi, sedangkan host yang berada pada VLAN berbeda tidak dapat berkomunikasi dan menghasilkan status host not reachable. Hasil tersebut membuktikan bahwa tag VLAN berbasis IEEE 802.1Q berhasil diterapkan pada setiap port Open vSwitch yang dikonfigurasi melalui Python script. Selain itu, keberhasilan pengujian juga menunjukkan bahwa GNS3 dapat digunakan sebagai lingkungan yang representatif untuk implementasi dan pengujian otomatisasi jaringan berbasis Open vSwitch, sebagaimana telah dibuktikan oleh Syah et al. (2020) dan Manik et al. (2023).

Jika dibandingkan dengan penelitian terdahulu, hasil penelitian ini memiliki kesamaan dengan penelitian Firmansyah (2024) dan Syah et al. (2020) yang menunjukkan bahwa network automation dapat meningkatkan efisiensi proses konfigurasi jaringan. Namun, kedua penelitian tersebut tidak secara khusus membahas otomatisasi konfigurasi VLAN pada Open vSwitch. Penelitian ini juga mendukung hasil penelitian Mauboy dan Wellem (2022) terkait efektivitas penggunaan Netmiko untuk otomatisasi perangkat jaringan, serta memperkuat penelitian Marzuki et al. (2023) yang mengotomatiskan Open vSwitch menggunakan Ansible. Perbedaannya, penelitian ini menggunakan Python scripting berbasis Netmiko pada lingkungan GNS3 dengan fokus khusus pada otomatisasi konfigurasi VLAN. Pendekatan yang diusulkan tidak memerlukan perangkat lunak tambahan selain layanan SSH pada Open vSwitch, sehingga lebih ringan dan mudah diterapkan. Dengan demikian, penelitian ini

memberikan kontribusi praktis berupa mekanisme otomatisasi konfigurasi VLAN pada Open vSwitch yang dapat membantu meningkatkan efisiensi, konsistensi, dan kemudahan pengelolaan jaringan virtual.

KESIMPULAN

Penelitian ini berhasil membuat sistem otomatisasi konfigurasi VLAN pada Open vSwitch menggunakan Python scripting dalam lingkungan simulasi GNS3. Hasil penelitian menunjukkan bahwa python script dengan library Netmiko untuk koneksi SSH dan dijalankan di komputer host/administrator mampu otomatis mengkonfigurasi bridge, port, dan tag VLAN pada Open vSwitch secara penuh hanya dengan satu kali eksekusi script. Sistem berhasil menjalankan setiap fungsinya sesuai dengan spesifikasi yang dirancang, termasuk pembuatan bridge, konfigurasi VLAN, verifikasi konfigurasi, komunikasi intra-VLAN, dan isolasi antar-VLAN. Selain itu, penelitian ini menghasilkan mekanisme otomatisasi konfigurasi VLAN pada Open vSwitch berbasis Python scripting dan Netmiko yang memberikan alternatif yang ringan, mudah diterapkan, dan mampu meningkatkan efisiensi pengelolaan jaringan virtual. Penelitian ini masih memiliki beberapa keterbatasan seperti masih dilakukan pada lingkungan simulasi GNS3 bukan lingkungan nyata dan dengan jumlah perangkat terbatas karena keterbatasan perangkat penelitian, serta belum dapat melakukan konfigurasi pada trunk port antar switch dan otomatis melakukan pemulihan konfigurasi(rollback). Oleh karena itu, untuk penelitian selanjutnya disarankan melakukan pada kondisi yang lebih mendekati praktik nyata, penambahan otomatisasi konfigurasi trunk port dan pemulihan konfigurasi (rollback), pengembangan antarmuka manajemen berbasis web, serta meningkatkan keandalan sistem dalam pengelolaan jaringan skala besar.

DAFTAR PUSTAKA

- Achmad Fauzi, Desi, A. K. S. (2026). Network Automation Dengan Metode Ndlc Menggunakan Bahasa Pemrograman Python. *Jurnal Riset Dan Aplikasi Mahasiswa Informatika (JRAMI)*, 07(01), 149–156.
- Dimas Ardiansyah. (2025). Rancang Bangun Sistem Monitoring Menggunakan Network Automation Netmiko Python. *Media Teknologi Dan Informatika*, 2, 140–148.
- Iqbal Firmansyah. (2024). Implementasi Network Automation Pada GNS3 Untuk Otomasi Konfigurasi Di PT.Semut Merah Squad. *Buletin Ilmiah Informatika Teknologi*, 2(2), 43–48.
- Kiswanto, D., Syahputra, H., & Panggabean, S. (2023). Training Peningkatan Kompetensi Industri Untuk Sertifikasi Profesi Network Engineer Skema Network+ Bersama Pt. Nusanet Dan Pt. Wilearning Indonesia. *Jurnal Umum Pengabdian Masyarakat (Jupemas)*, 2(1), 43–49.
- Lukas Amantha Olan Sahat Manik, Ir. Theresia Ghozali, M.Sc, V. W., & Mahyastuty. (2023). Perancangan Jaringan Internet Menggunakan GNS3, Qemu, dan Virtual Box. *Jurnal Elektro*, 15(1), 10.
- Marzuki, K., Kholid, M. I., Hariyadi, I. P., & Mardedi, L. Z. A. (2023). Automation of Open VSwitch-Based Virtual Network Configuration Using Ansible on Proxmox Virtual Environment. *International Journal of Electronics and Communications Systems*, 3(1), 11–20. <https://doi.org/10.24042/ijecs.v3i1.16524>
- Mauboy, L. G., & Wellem, T. (2022). Studi Perbandingan Library Untuk Implementasi Network Automation Menggunakan Paramiko Dan Netmiko Pada Router Mikrotik. *JURIKOM (Jurnal Riset Komputer)*, 9(4), 790–799. <https://doi.org/10.30865/jurikom.v9i4.4420>
- Mazin, A. A., Abidin, H. Z., Mazalan, L., & Mazin, A. M. (2023). Network Automation Using Python Programming to Interact with Multiple Third-Party Network Devices. *2023 10th*



International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), 59–64. <https://doi.org/10.1109/ICITACEE58587.2023.10277400>

Mazin, A. M., Rahman, R. A., Kassim, M., & Mahmud, A. R. (2021). Performance Analysis on Network Automation Interaction with Network Devices Using Python. *2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 360–366. <https://doi.org/10.1109/ISCAIE51753.2021.9431823>

Rahardika, D., & Ratama, N. (2021). Implementasi Network Automation Untuk Konfigurasi Jaringan Baru Dengan Netmiko. *Journal of Artificial Intelligence and Innovative Applications*, 2(3), 190–200.

Syah, I., Muhammad, A. H., & Gunawan, E. (2020). Simulasi Network Automation Menggunakan Ansible Di GNS3 (Studi Kasus Smile Project). *Jurnal Teknik Informatika (J-Tifa)*, 3(2), 1–8. <https://doi.org/10.52046/j-tifa.v3i2.1065>

Yalestia Chandrawaty, N. M. A., & Hariyadi, I. P. (2021). Implementasi Ansible Playbook Untuk Mengotomatisasi Manajemen Konfigurasi VLAN Berbasis VTP Dan Layanan DHCP. *Jurnal Bumigora Information Technology (BITE)*, 3(2), 107–122. <https://doi.org/10.30812/bite.v3i2.1577>

Yuansa Alfaresa, Bongga Arifwidodo, F. K. (2023). Automate IGP And EGP Routing Protocol Configuration Using A Network Automation Library. *JOIN (Jurnal Online Informatika)*, 8(2), 222–231. <https://doi.org/10.15575/join.v8i2.1157>