

Otomatisasi Management Konfigurasi VPLS Menggunakan Ansible

Haris Noegroho, Husain, Lilik Widyawati

Universitas Bumigora, Mataram, Indonesia

Correspondence : e-mail: harissrewok@gmail.com

Abstrak

Perkembangan teknologi jaringan menuntut efisiensi dan akurasi dalam proses konfigurasi, khususnya pada Virtual Private LAN Service (VPLS) yang memiliki kompleksitas tinggi dan rentan terhadap kesalahan manual. Penelitian ini bertujuan mengimplementasikan otomatisasi manajemen konfigurasi VPLS pada perangkat MikroTik menggunakan Ansible, sebuah alat otomatisasi yang bersifat agentless dan memanfaatkan protokol SSH. Metodologi yang digunakan adalah Network Development Life Cycle (NDLC) dengan tiga tahap utama: analisis, desain, serta simulasi dan prototyping. Implementasi dilakukan pada lingkungan simulasi GNS3 dengan tiga router MikroTik dan satu server Ansible berbasis Ubuntu melalui WSL. Pengujian membandingkan konfigurasi manual dengan otomatisasi dalam hal waktu implementasi, konsistensi hasil, dan kinerja jaringan. Hasil menunjukkan bahwa kedua metode memberikan konektivitas stabil dengan latensi rendah (1–3 ms) dan tanpa packet loss, namun otomatisasi secara signifikan mempercepat proses konfigurasi serta meminimalkan human error. Temuan ini menegaskan bahwa penerapan Ansible dalam konfigurasi VPLS meningkatkan efisiensi, konsistensi, dan skalabilitas manajemen jaringan, sehingga layak diadopsi pada infrastruktur berskala besar.

Kata kunci: Ansible, Otomatisasi Jaringan, Manajemen Konfigurasi, Virtual Private LAN Service (VPLS), Multi Protocol Label Switching (MPLS).

Abstract

The rapid development of networking technology demands efficiency and accuracy in configuration processes, particularly for Virtual Private LAN Service (VPLS), which is highly complex and prone to human error. This research aims to implement automated VPLS configuration management on MikroTik devices using Ansible, an agentless automation tool that utilizes the SSH protocol. The methodology applied is the Network Development Life Cycle (NDLC) with three main stages: analysis, design, and simulation & prototyping. The implementation was carried out in a GNS3 simulation environment with three MikroTik routers and one Ansible server running on Ubuntu via WSL. Testing compared manual configuration with automation in terms of deployment time, result consistency, and network performance. The results show that both methods provided stable connectivity with low latency (1–3 ms) and zero packet loss. However, automation significantly reduced configuration time and minimized human error. These findings highlight that applying Ansible for VPLS configuration enhances efficiency, consistency, and scalability in network management, making it highly suitable for large-scale infrastructure adoption.

Keywords: Ansible, Network automation.

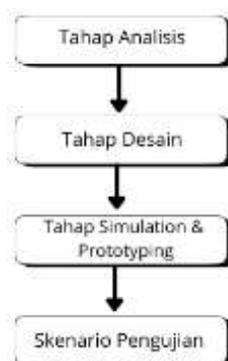
1. Pendahuluan

Perkembangan ilmu pengetahuan dan teknologi yang pesat membuat teknologi digunakan hampir di seluruh aspek kehidupan. Salah satu dampak positifnya adalah kemudahan manusia dalam memperoleh informasi dan mengotomatisasi pekerjaan, sehingga waktu pengerjaan menjadi lebih efisien. Salah satu bentuk perkembangan teknologi tersebut adalah virtualisasi, yaitu proses menciptakan bentuk virtual dari suatu entitas fisik. VPLS merupakan teknologi layer dua yang menghubungkan beberapa situs melalui jaringan Internet Protocol (IP) yang dikelola penyedia layanan dengan dukungan Multi Protocol

Label Switching (MPLS). Teknologi ini memungkinkan perangkat jaringan yang berjauhan berkomunikasi seolah berada dalam jaringan lokal, sekaligus memaksimalkan bandwidth komunikasi. Namun, seiring meningkatnya jumlah pengguna VPLS, konfigurasi menjadi lebih kompleks dan rawan kesalahan, terutama pada skala besar. Kondisi ini mendorong penelitian untuk mengembangkan metode otomatisasi konfigurasi.

2. Metode Penelitian

Pada tahap ini metode yang digunakan adalah NDLC (*Network Development Life Cycle*) yang dapat dilihat pada Gambar 1, namun dari metode tersebut, hanya tiga tahap yang diambil, yaitu Analysis, Design, dan Simulation & Prototyping.



Gambar 1. Metode Penelitian

1. Tahap Analisis

Pada tahap analisis, peneliti melakukan pengumpulan data melalui studi literatur dengan cara membaca dan mempelajari berbagai jurnal ilmiah, artikel, dan skripsi yang membahas mengenai Virtual Private LAN Service (VPLS) serta teknologi pendukungnya, termasuk metode otomatisasi konfigurasi jaringan. Dari studi ini, peneliti memperoleh informasi mengenai konsep, cara kerja, dan penerapan VPLS pada berbagai skenario, baik di jaringan tradisional maupun berbasis Software Defined Network (SDN), serta integrasinya dengan fitur seperti Traffic Engineering (TE) Tunnel. Data yang terkumpul kemudian dianalisis untuk mengidentifikasi kelebihan dan kekurangan dari metode konfigurasi yang ada, sehingga peneliti dapat menentukan bahwa penggunaan Ansible sebagai alat otomatisasi dipilih karena mampu mempercepat proses konfigurasi, menjaga konsistensi pengaturan, dan meminimalkan kesalahan manusia (*human error*) dalam penerapan VPLS.

2. Tahap Desain

Pada tahap desain, peneliti membuat rancangan implementasi sistem otomatisasi konfigurasi VPLS menggunakan Ansible. Rancangan ini dimulai dengan membuat topologi uji coba jaringan di GNS3 yang terdiri dari tiga router MikroTik, di mana Router1 berfungsi sebagai gateway ke internet, sedangkan Router2 dan Router3 berperan sebagai endpoint lokal untuk masing-masing kelompok pengguna. Selanjutnya, peneliti menentukan rancangan pengalamatan IP untuk setiap perangkat dan interface sesuai dengan peran masing-masing, termasuk koneksi point-to-point antar router. Setelah itu, peneliti menetapkan kebutuhan perangkat keras dan perangkat lunak yang digunakan, seperti laptop dengan spesifikasi tertentu, sistem operasi Windows 11 dengan WSL2 Ubuntu 22.04, GNS3, MikroTik CHR, serta Ansible beserta modul-modul pendukungnya. Terakhir, peneliti menyusun rancangan sistem otomatisasi berupa flowchart yang menggambarkan alur proses konfigurasi, mulai dari mengakses router melalui SSH, menetapkan alamat IP, mengaktifkan MPLS, membuat interface VPLS, menggabungkannya ke bridge local, hingga tahap verifikasi konfigurasi.

3. Tahap Simulation dan Prototyping

Pada tahap simulation & prototyping, peneliti mengimplementasikan rancangan sistem yang telah dibuat pada lingkungan simulasi menggunakan GNS3. Proses ini diawali dengan membangun topologi jaringan yang terdiri dari tiga router MikroTik dan satu server Ansible yang dijalankan di Ubuntu melalui WSL2 pada sistem operasi Windows. Selanjutnya dilakukan instalasi Ansible beserta modul-modul pendukungnya, pembuatan *inventory file* berisi daftar alamat IP router yang akan dikonfigurasi, serta penulisan *playbook* Ansible dalam format YAML yang berisi perintah otomatisasi konfigurasi VPLS. Setelah semua persiapan selesai, *playbook* dijalankan untuk mengonfigurasi seluruh router secara otomatis sesuai skenario yang telah dirancang. Tahap ini diakhiri dengan proses verifikasi dan pengujian, yaitu memeriksa konfigurasi pada masing-masing router melalui CLI MikroTik serta

melakukan *ping test* untuk memastikan konektivitas antar perangkat dan keberhasilan pembentukan jaringan VPLS.

4. Tahap Skenario Pengujian

Pada tahap skenario pengujian, peneliti melakukan serangkaian tes untuk mengevaluasi efektivitas dan keberhasilan sistem otomatisasi konfigurasi VPLS yang telah dibuat menggunakan Ansible. Pengujian dilakukan dengan mengamati beberapa aspek utama, yaitu keberhasilan pembentukan koneksi VPLS antar lokasi, kemampuan perangkat untuk saling terhubung melalui jaringan meskipun berada pada LAN yang berbeda, efisiensi waktu konfigurasi dibandingkan metode manual, serta konsistensi hasil konfigurasi di seluruh perangkat. Selain itu, peneliti juga menguji toleransi sistem terhadap kemungkinan kesalahan atau *error* selama proses otomatisasi, dengan tujuan memastikan bahwa konfigurasi tetap dapat berjalan dengan baik dan stabil meskipun terdapat gangguan pada salah satu bagian proses.

3. Hasil dan Pembahasan

Hasil implementasi konfigurasi VPLS pada jaringan MikroTik, yang dilakukan dengan dua metode, yaitu konfigurasi manual dan otomatisasi menggunakan Ansible. Konfigurasi Manual merupakan proses pengaturan jaringan VPLS yang dilakukan langsung pada setiap router MikroTik melalui antarmuka CLI tanpa bantuan sistem otomatisasi. Proses ini membutuhkan interaksi langsung administrator jaringan dengan perangkat, di mana setiap perintah diketik dan dijalankan satu per satu dengan tahapan berikut :

1. Konfigurasi IP Adress

Pada tahap konfigurasi IP address, setiap interface pada router diatur sesuai dengan rancangan topologi jaringan yang telah dibuat. Router1 memiliki tiga interface, yaitu ether1 yang terhubung ke internet dengan alamat IP 192.168.0.2/24, ether2 yang terhubung ke Router2 dengan alamat IP 10.10.10.1/30, dan ether3 yang terhubung ke Router3 dengan alamat IP 10.10.10.5/30. Router2 dikonfigurasi dengan ether1 yang terhubung ke Router1 menggunakan alamat IP 10.10.10.2/30, serta bridge-local dengan alamat IP 192.168.10.1/24 sebagai jaringan lokal untuk User1 dan User2. Sementara itu, Router3 memiliki ether1 yang terhubung ke Router1 dengan alamat IP 10.10.10.6/30, dan bridge-local dengan alamat IP 192.168.20.1/24 untuk jaringan lokal User3 dan User4. Pengaturan ini memastikan setiap router memiliki koneksi yang jelas antar perangkat dan memudahkan proses komunikasi jaringan.

2. Pembuatan Bridge Lokal

Pada tahap pembuatan bridge lokal, Router2 dan Router3 dikonfigurasi untuk membuat sebuah interface bridge yang berfungsi menghubungkan port fisik yang terhubung ke perangkat pengguna dengan jaringan lokal masing-masing router. Di Router2, bridge-local menghubungkan port yang digunakan oleh User1 dan User2, sedangkan di Router3, bridge-local menghubungkan port yang digunakan oleh User3 dan User4. Dengan adanya bridge lokal ini, seluruh perangkat pengguna di masing-masing router dapat saling terhubung dan berkomunikasi dalam satu segmen jaringan yang sama, sehingga mempermudah pengelolaan trafik dan distribusi data di dalam jaringan lokal.

3. Aktivasi MPLS

Pada tahap aktivasi MPLS, setiap router dikonfigurasi untuk mengaktifkan fitur *Multiprotocol Label Switching* pada interface yang terhubung antar-router. MPLS ini digunakan untuk membawa label khusus pada paket data sehingga memungkinkan pembentukan koneksi Layer 2 antar lokasi secara efisien. Pada Router1, MPLS diaktifkan pada interface ether2 yang terhubung ke Router2 dan ether3 yang terhubung ke Router3, sementara pada Router2 dan Router3, MPLS diaktifkan pada interface yang terhubung langsung ke Router1. Dengan pengaturan ini, setiap link antar-router dapat mengirimkan paket data berlabel yang dibutuhkan untuk membentuk koneksi VPLS, sehingga jaringan dapat beroperasi seperti berada dalam satu LAN meskipun terpisah secara geografis.

4. Pembuatan Interface VPLS

Pada tahap pembuatan interface VPLS, setiap router dikonfigurasi untuk membuat antarmuka Virtual Private LAN Service yang akan menjadi jalur komunikasi Layer 2 antar lokasi. Router1 memiliki dua interface VPLS, yaitu *vpls-r1-r2* untuk menghubungkan Router1 dengan Router2 menggunakan *remote-address* 10.10.10.2, dan *vpls-r1-r3* untuk menghubungkan Router1 dengan Router3 menggunakan *remote-address* 10.10.10.6. disisi lain, Router2 membuat interface *vpls-r2-r1* yang mengarah ke Router1, sedangkan Router3 membuat interface *vpls-r3-r1* yang juga mengarah ke Router1. Setiap interface VPLS diberi identitas unik dan *vpls-id* yang sama di kedua ujung koneksi, sehingga kedua router dapat mengenali bahwa mereka berada dalam satu tunnel yang sama. Pengaturan ini memastikan bahwa lalu lintas data antar lokasi dapat berjalan seolah-olah semua perangkat berada dalam satu jaringan lokal.

5. Penggabungan VPLS dengan Bridge Lokal

Pada tahap penggabungan VPLS dengan bridge lokal, interface VPLS yang telah dibuat pada Router2 dan Router3 dihubungkan ke bridge lokal masing-masing. Proses ini bertujuan agar lalu lintas data dari perangkat pengguna yang terhubung ke bridge lokal dapat langsung dikirim melalui koneksi VPLS menuju router di lokasi lain, sehingga seluruh perangkat pada kedua sisi dapat berkomunikasi seolah berada dalam satu LAN fisik. Sebagai contoh, di Router2, interface *vpls-r2-r1* digabungkan dengan bridge-lokal yang menghubungkan User1 dan User2, sementara di Router3, interface *vpls-r3-r1* digabungkan dengan bridge lokal yang menghubungkan User3 dan User4. Dengan cara ini, konektivitas antar pengguna di lokasi berbeda dapat tercapai secara transparan tanpa memerlukan konfigurasi tambahan pada sisi pengguna.

6. Verifikasi Konfigurasi

Pada tahap verifikasi konfigurasi, administrator memeriksa seluruh pengaturan yang telah dilakukan pada masing-masing router untuk memastikan bahwa konfigurasi VPLS telah dibuat dengan benar dan sesuai rancangan. Proses verifikasi ini dilakukan menggunakan perintah *print* pada Command Line Interface (CLI) MikroTik, yang menampilkan detail konfigurasi seperti alamat IP, pengaturan bridge, status MPLS, interface VPLS, serta port yang terhubung. Dengan verifikasi ini, dapat dipastikan bahwa semua parameter telah diatur secara konsisten pada setiap router, sehingga jaringan VPLS siap diuji tanpa adanya kesalahan konfigurasi yang dapat mengganggu konektivitas.

7. Pengujian Konektivitas

Pada tahap pengujian konektivitas, dilakukan pengujian menggunakan perintah *ping* dari masing-masing perangkat pengguna untuk memastikan bahwa koneksi antar lokasi dalam jaringan VPLS berfungsi dengan baik. Pengujian ini mencakup komunikasi antar user yang berada pada router berbeda serta koneksi ke internet melalui Router1. Hasil pengujian menunjukkan bahwa seluruh perangkat dapat saling terhubung dengan stabil, ditandai dengan waktu *latency* rata-rata 15–18 milidetik dan tanpa terjadinya *packet loss*. Hal ini membuktikan bahwa konfigurasi manual VPLS telah berhasil dijalankan dengan baik dan mampu menyediakan koneksi yang andal antar lokasi seolah berada pada satu jaringan lokal yang sama.

```
[user1] > ping 192.168.0.1 count=5
SEQUENCE TTL TIME(ms)
0      64  14
1      64  15
2      64  14
3      64  15
4      64  16

5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 14/14.8/16 ms
```

Gambar 2. Ping User 1 ke Cloud

Pada metode otomatisasi, proses konfigurasi VPLS dilakukan menggunakan Ansible, sebuah *configuration management tool* yang memungkinkan penulis mengatur seluruh router MikroTik secara terpusat hanya dari satu server. Server ini dijalankan pada Ubuntu 22.04 di dalam WSL2 pada sistem operasi Windows. Langkah awal adalah menyiapkan file *inventory* yang berisi daftar IP address dari semua router target (Router1, Router2, Router3) beserta kredensial login melalui SSH. File ini menjadi acuan Ansible untuk mengetahui perangkat mana saja yang akan dikonfigurasi. Selanjutnya, penulis membuat playbook Ansible dalam format YAML yang memuat seluruh instruksi konfigurasi jaringan. Playbook ini berisi beberapa tahap penting :

1. Pengaturan IP Address pada masing-masing interface router sesuai topologi jaringan.
2. Aktivasi MPLS pada interface yang digunakan sebagai jalur komunikasi antar-router.
3. Penambahan Interface MPLS untuk membawa label MPLS yang dibutuhkan oleh VPLS.
4. Pembuatan Interface VPLS di setiap router dengan pengaturan *vpls-id* yang sama dan *site-id* berbeda, agar dapat saling terhubung pada layer 2.
5. Penggabungan Interface VPLS dan Bridge Lokal sehingga trafik dari pengguna di LAN masing-masing router dapat melewati koneksi VPLS seolah berada dalam satu jaringan local

Proses otomatisasi dimulai dengan menjalankan perintah `ansible-playbook vpls.yml -i inventory.ini`, yang secara otomatis mengirimkan konfigurasi ke seluruh router sesuai peran masing-masing dalam topologi, tanpa perlu melakukan pengaturan manual satu per satu. Setelah eksekusi selesai, dilakukan verifikasi melalui CLI MikroTik untuk memastikan setiap langkah konfigurasi — mulai dari pengaturan IP, aktivasi MPLS, pembuatan interface VPLS, hingga penggabungan ke bridge lokal — telah berhasil. Selanjutnya, pengujian konektivitas dilakukan menggunakan *ping test* antar-user di Router2 dan Router3, serta ke internet melalui Router1. Hasilnya menunjukkan koneksi stabil dengan *latency* rendah dan tanpa *packet loss*. Dibandingkan metode manual, pendekatan otomatisasi ini memberikan kecepatan konfigurasi yang jauh lebih tinggi, hasil yang konsisten di semua perangkat, serta meminimalkan risiko *human error*.

```
/ip address print
0 ether1 10.10.10.6/30
1 bridge-local 192.168.20.1/24

/routing mpls interface print
0 ether1

/interface vpls print
0 vpls1 vpls-id=100 site-id=3 interface=ether1

/interface bridge print
0 bridge-vpls
1 bridge-local

/interface bridge port print
0 vpls1 bridge-vpls
1 bridge-local bridge-vpls
```

Gambar 3. Hasil Print Konfigurasi Playbook

```
[user1] > ping 192.168.0.1 count=5

SEQUENCE TTL TIME(ms)
0      64  15
1      64  14
2      64  15
3      64  16
4      64  14

5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 14/14.8/16 ms
```

Gambar 4. Hasil Ping User1 ke Cloud

Selanjutnya pengujian dilakukan dengan metode yang sama untuk kedua pendekatan, yaitu *ping test* antar user pada jaringan VPLS dan ke jaringan internet. Hasilnya menunjukkan bahwa baik konfigurasi manual maupun otomatisasi sama-sama berhasil membentuk konektivitas end-to-end dengan latency rata-rata 15–18 ms dan 0% packet loss, menandakan kualitas koneksi yang stabil. Perbedaan dapat dilihat pada Tabel 1 dan pada waktu implementasi, kemudahan pengelolaan, dan konsistensi konfigurasi :

1. Waktu Implementasi: Manual memakan waktu lama karena konfigurasi dilakukan per perangkat; otomatisasi lebih cepat karena cukup menjalankan satu *playbook* untuk semua perangkat.
2. Kemudahan Pengelolaan: Manual memerlukan pengulangan di tiap perangkat; otomatisasi cukup mengubah satu file dan semua perangkat diperbarui otomatis.
3. Konsistensi Hasil: Manual rentan *human error*; otomatisasi memberikan konfigurasi yang seragam di seluruh jaringan.

Tabel 1 Analisis Hasil Uji Coba

Dari	Ke	Manual (ms)	Otomatisasi (ms)
User 1	Cloud	15	15
User 1	User 3	18	18
User 2	Cloud	16	15
User 2	User 4	18	18
User 3	Cloud	17	16
User 3	User 1	18	18
User 4	Cloud	16	16

4. Kesimpulan

Penelitian ini membuktikan bahwa otomatisasi konfigurasi Virtual Private LAN Service (VPLS) menggunakan Ansible dapat menghasilkan performa jaringan yang setara dengan konfigurasi manual, namun dengan keunggulan yang signifikan dalam efisiensi waktu, kemudahan pengelolaan, dan konsistensi hasil. Kedua metode sama-sama mampu membentuk koneksi VPLS yang stabil dengan latency rata-rata 15–18 ms dan 0% *packet loss*. Metode manual membutuhkan waktu lebih lama dan rentan terhadap kesalahan manusia karena konfigurasi dilakukan satu per satu pada setiap perangkat. Sebaliknya, otomatisasi memungkinkan penerapan konfigurasi ke banyak perangkat secara serentak hanya dengan satu perintah *playbook*, meminimalkan risiko kesalahan dan memastikan konfigurasi yang seragam di seluruh jaringan. Secara keseluruhan, penggunaan Ansible dalam konfigurasi VPLS sangat direkomendasikan, terutama untuk jaringan berskala besar, karena mampu meningkatkan produktivitas, mengurangi potensi *human error*, dan memudahkan pemeliharaan jaringan di masa depan.

5. Ucapan Terima

Kami ingin mengucapkan terima kasih kepada Bapak Husain sebagai pembimbing pertama dan Ibu Lilik Widyawati sebagai pembimbing kedua atas bimbingan dan dukungan mereka selama penelitian ini.

6. Daftar Pustaka

- [1] Angsori, M. L. (n.d.). *Jaringan komputer*.
- [2] Ardhiansyah, M., Noris, S., & Andrianto, R. (n.d.). *Jaringan komputer* (Issue 1).
- [3] Attari, A., Negara, R. M., & Sanjoyo, D. D. (2019). Analisis Performansi High Availability Jaringan pada Virtual Private LAN Service Legasi dan Berbasis Software Defined Network. *Prosiding SENIATI*, 5(3), 99–104.
- [4] Cerdas Kurniawan, R. S., Tulloh, R., & Dyah Irawai, I. S. (2020). Implementasi Vpls Pada Jaringan Software Defined Network (Sdn) Dengan Menggunakan Onos Controller Berbasis Raspberry-Pi 3 Implementation of Vpls on Software Defined Network Using Onos Controller Based on Raspberry-Pi 3. 6(2), 3796–3805.
- [5] Essy, I., & Sari, M. (n.d.). *Jaringan Komputer*.
- [6] Evy Nurmiati. (2012). Analisis Dan Perancangan Web Server Pada Handphone. *Web Server Handphone*, 5(2), 1–17.
- [7] Hidayanto, F., & Zidni Ilimi, M. (2015). Pentingnya Internet Sehat. *Jurnal Inovasi Dan Kewirausahaan*, 4(1), 21–24. <https://jurnal.uui.ac.id/ajie/article/view/7888%0Ahttps://journal.uui.ac.id/ajie/article/download/7888/6897>
- [8] Iqromuddin, F., Nathasia, N. D., Fitri, I., & Nasional, U. (2017). *Peningkatan Konektivitas Service VPLS Redundant Path Dengan Rapid Spanning Tree Protocol*. 2(2), 47–52. <https://doi.org/10.31328/jo>
- [9] Munadi, R., & Mayasari, R. (2016). IMPLEMENTASI DAN ANALISIS PERFORMANSI JARINGAN VIRTUAL PRIVATE LAN SERVICE-TE TUNNEL DENGAN OPENIMSCORE

SEBAGAI SERVER LAYANAN MULTIMEDIA Implementation and Performance Analysis of Virtual Private LAN Service-TE Tunnel Network with OpenIMSCore as Multimed. 3(3), 4641–4648.

- [10] Novanto, M. R., Dedy Irawan, J., & Ariwibisono, F. X. (2022). Rancang Bangun Panel Virtual Private Network (Vpn) Berbasis Web. *JsATI (Jurnal Mahasiswa Teknik Informatika)*, 6(1), 259–266. <https://doi.org/10.36040/jati.v6i1.4615>
- [11] Purbo, O. W. (2018). *Internet – Tcp / Ip* : 264