Otomatisasi Manajemen *Virtual Private Network* (VPN) Berbasis *OpenVPN* Pada *Mikrotik* Menggunakan *Ansible*

Yogi Eka Putrawan, I Putu Hariyadi, Husain, Lilik Widyawati, Kurniadin Abd Latif
Universitas Bumigora, Mataram, Indonesia

Correspondence: *e-mail*: ogik8556@gmail.com

ABSTRAK

Perkembangan teknologi jaringan menuntut efisiensi dan keandalan dalam pengelolaan infrastruktur VPN, terutama dalam lingkungan perusahaan dan organisasi yang kompleks. Penelitian ini bertujuan untuk mengotomatisasi proses konfigurasi OpenVPN pada perangkat MikroTik menggunakan Ansible Playbook guna meningkatkan efisiensi, konsistensi, dan mengurangi risiko human error. Metodologi yang digunakan meliputi analisis kebutuhan, perancangan sistem, implementasi otomatisasi konfigurasi, serta pengujian pada jaringan skala kecil hingga menengah. Hasil penelitian menunjukkan bahwa otomatisasi konfigurasi menggunakan Ansible mampu memangkas waktu provisioning dari sekitar 10 menit menjadi kurang dari 1 menit per perangkat, serta menjamin keseragaman konfigurasi di seluruh perangkat jaringan. Selain itu, penerapan otomatisasi ini dapat memperkuat aspek keamanan dan memudahkan proses pemantauan jaringan VPN secara otomatis melalui sistem monitoring terintegrasi. Berdasarkan hasil tersebut, disarankan agar sistem otomasi ini dikembangkan lebih lanjut dengan fitur monitoring, manajemen akses berbasis peran, dan dokumentasi otomatis untuk mendukung pengelolaan jaringan yang lebih efisien dan aman di masa mendatang.

Kata kunci: hoaks politik, TF-IDF, word embedding, klasifikasi, machine learning.

ABSTRACT

The development of network technology demands efficiency and reliability in the management of VPN infrastructure, especially in complex corporate and organizational environments. This research aims to automate the OpenVPN configuration process on MikroTik devices using Ansible Playbook to improve efficiency, consistency, and reduce the risk of human error. The methodology used includes requirements analysis, system design, configuration automation implementation, and testing on small to medium scale networks. The results show that configuration automation using Ansible can cut provisioning time from about 10 minutes to less than 1 minute per device, and ensure configuration uniformity across network devices. In addition, the implementation of this automation can strengthen the security aspect and facilitate the process of monitoring the VPN network automatically through an integrated monitoring system. Based on these results, it is recommended that this automation system be further developed with monitoring features, role-based access management, and automated documentation to support more efficient and secure network management in the future.
Key words: political hoax, TF-IDF, word embedding, classification, machine learning.

1. PENDAHULUAN

Di era digital yang berkembang pesat ini sehingga meningkatnya perkembangan teknologi saat ini akan berdampak pula pada kebutuhan jaringan yang digunakan. Jaringan komputer adalah gabungan dari dua komputer atau lebih yang telah didesain sedemikian rupa agar dapat saling terhubung satu sama lain untuk dapat melakukan komunikasi, berbagi sumber daya maupun berbagi informasi[1]. Informasi dan data bergerak melalui kabel-kabel atau tanpa kabel sehingga memungkinkan pengguna jaringan komputer dapat saling bertukar dokumen dan data, mencetak pada printer yang sama dan bersama-sama menggunakan hardware atau software yang terhubung. Sama halnya pada jaringan *Wide Area Network* (WAN)[2]. WAN adalah tipe jaringan komputer yang mencakup area geografi yang luas, seperti Kota, Negara, ataupun Benua. WAN digunakan untuk menghubungkan beberapa *Local Area Network* (LAN) dan *Metropolitan Area Network (*MAN) sehingga perangkat yang berada di lokasi yang jauh dapat saling berkomunikasi, oleh karena itu keamanan jaringan menjadi salah satu aspek yang sangat penting untuk diperhatikan. Perkembangan teknologi informasi dan komunikasi telah memberikan banyak kemudahan bagi kehidupan kita, sehingga menghadirkan risiko serius terkait keamanan jaringan ini[3][2]. Beberapa masalah yang sering ditemui yaitu update pada suatu software maka otomatis setiap *server* tersebut harus di-update satu persatu, Apabila ada lebih dari satu dan konfigurasi *Network* tersebut sama persis dan diharuskan mengkonfigurasi hal tersebut satu persatu, dan masih banyak lagi. Pengelolaan infrastruktur jaringan menggunakan penerapan manual dapat mengakibatkan biaya operasional meningkat, tidak efisien dan rentan terhadap kesalahan, dan memerlukan waktu yang di butuhkan cukup lama seiring dengan bertambahnya perangkat jaringan yang akan dikelola[4].

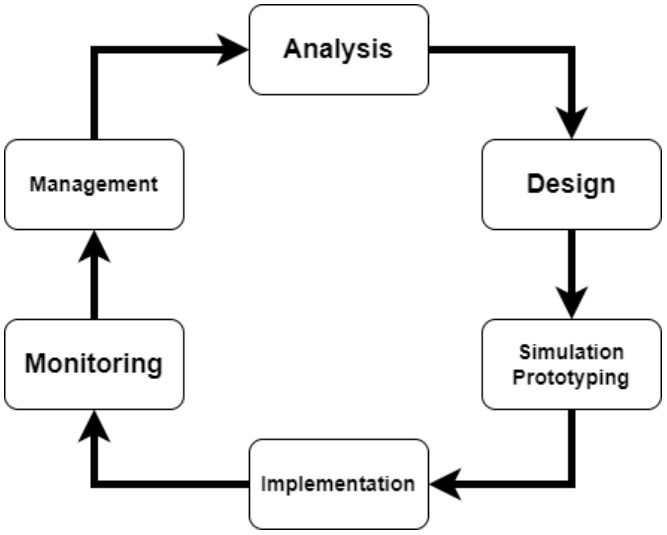
Penelitian ini akan menerpakan otomatisasi manajemen *Virtual Private network (*VPN) dengan menggunakan *Protocol OpenVPN* . Otomatisasi jaringan merupakan sebuah solusi dalam mengimplementasikan beberapa pekerjaan yang rumit atau berulang. Dalam metode tradisional konfigurasi pada perangkat jaringan dilakukan dengan masuk ke dalam perangkat jaringan untuk mengkonfigurasi perangkat jaringan tersebut, pekerjaan tersebut terlihat mudah bila yang dikonfigurasi masih satu atau dua perangkat jaringan, hal ini akan terlihat rumit jika terdapat banyak perangkat jaringan seperti *server* yang harus dikonfigurasi[5][6][7][8]. Oleh karena itu pada penelitian ini dilakukan otomatisasi dengan menerapkan Open VPN menggunakan *Ansible playbook* untuk mengatasi permasalahan tersebut. *Ansible* merupakan tool bersifat open source yang bertugas sebagai alat otomatisasi jaringan pada skala besar dan kecil pada sebuah jaringan komputer serta *Ansible* bertugas untuk mempermudah pekerjaan

dalam mengkonfigurasi perangkat jaringan[9][10][6][1][11]. Sedangkan VPN merupakan salah satu teknologi yang banyak digunakan dalam membangun jaringan pribadi virtual. Teknologi ini menghubungkan lokasi yang jauh secara *geografis* dengan aman dan efektif seolah-olah mereka adalah bagian dari jaringan lokal yang Sama[12][13][14].

Mempersiapan Infrastruktur Identifikasi Perangkat jaringan yang akan dikelola, seperti server, internet, komputer. Lalu melakukan instalasi dan konfigurasi *Open Virtual Private Network* (OVPN), dan inventory ansibel untuk memncantumkan semua perangkat yang akan di kelola untuk dilakukan Penerapan konfigurasi jaringan OVPN secara otomatis untuk mengurangi tingkat kesalahan. Pada penelitian ini digunakan metode penelitian NDLC (*Network* Development Life Cycle) metode ini merujuk pada Analisis, Desain, Simulasi, Implementasi, monitoring, Management Otomatisasi Manajemen *Open Virtual Private Network* (OVPN) menggunakan *Ansible Playbook* yang menampung kebijakan konfigurasi sehingga dapat mengotomatisasi manajemen OVPN secara dinamis. Sistem otomatisasi yang dibangun dapat memanajemen konfigurasi perangkat jaringan baik server, internet, data, dan komputer meliputi pembuatan dan penghapusan konfigurasi OVPN dengan otomatisasi menggunakan ansibel *playbook* di setiap lokasi untuk lebih efisien dan meminimalisir kesalahan yang dilakukan secara manual.

2. METODE PENELITIAN

Metode penelitian yang digunakan pada penelitian ini adalah *Network* Development Life Cycle (NDLC). Menurut Goldman dan Rawles (2004), NDLC merupakan model dibalik kunci perancangan jaringan komputer. NDLC merupakan sebuah siklus proses perancangan atau pengembangan suatu infrastruktur jaringan yang memungkinkan terjadinya pemantauan jaringan untuk mengetahui statistik dan kinerja jaringan. NDLC juga mempunyai elemen yang mendefinisikan fase, tahapan, langkah atau mekanisme proses yang menggambarkan secara keseluruhan proses dan tahapan pengembagan sistem jaringan yang berkesinambungan. Adapun siklus dari tahapan metodologi NDLC yakni analysis, design, simulation prototyping, implementation, monitoring dan Management[15][4][1]. seperti terlihat pada gambar di bawah ini. pada penelitian ini hanya menggunakan tiga tahapan, metode NDLC digunakan agar penulis dapat memiliki urutan kerja yang efisien.



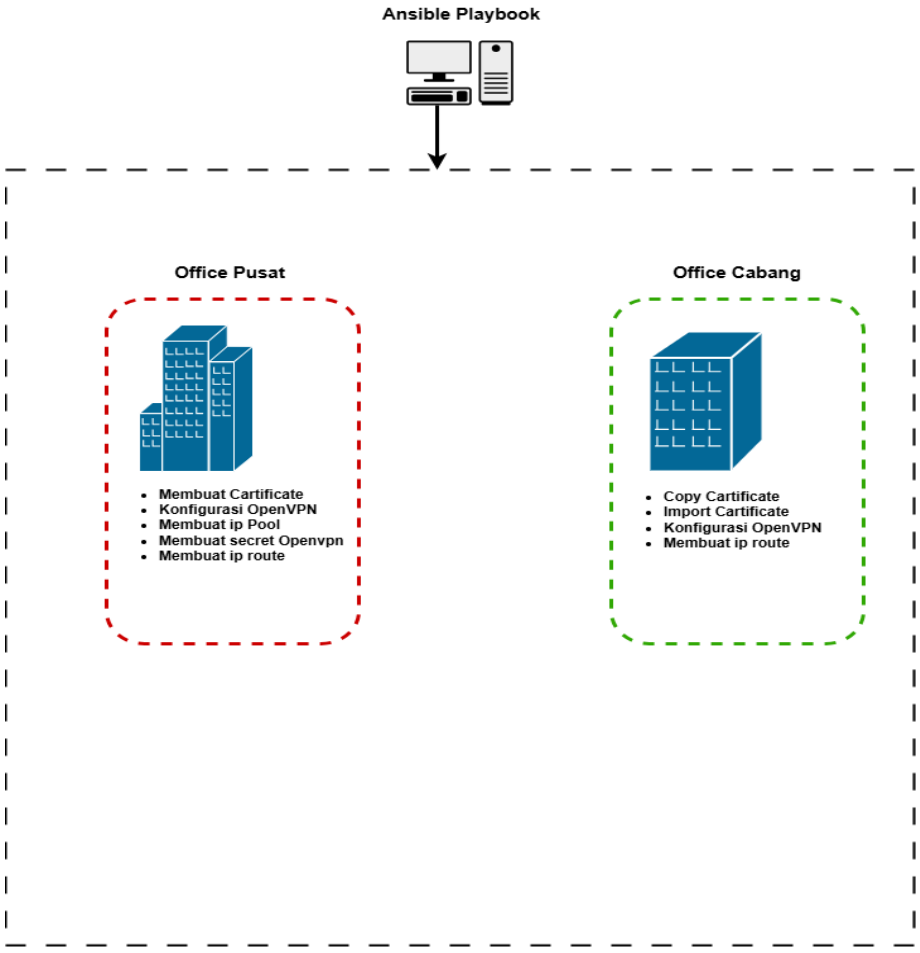
Gambar 1 Network Development Life Cycle

A. Tahap analisys

Pada tahap ini, akan dijelaskan proses analisis yang dilakukan dalam rangka mendukung pengembangan sistem otomatisasi manajemen *Virtual Private* network menggunakan ansible. Analisis ini terbagi menjadi dua bagian utama, yaitu pengumpulan data dan analisis data. Bagian pertama memaparkan hasil pengumpulan data berupa artikel penelitian terdahulu yang relevan, sementara bagian kedua menyajikan hasil analisis terhadap isi dan temuan dari penelitian tersebut, serta perbandingan dengan fokus dan solusi yang diusung dalam penelitian ini.

1) Pengumpulan data
   dilakukan pengumpulan data melalui studi literatur dan analisis data yang terkumpul. Pengumpulan data berupa artikel tentang manajemen konfigurasi *OpenVPN* pada infrastruktur *Wide Area Network* (WAN). Berdasarkan hasil dari analisis penelitian terdahulu yang telah dilakukan, dapat disimpulkan bahwa penelitian sebelumnya hanya berfokus pada manajemen konfigurasi *OpenVPN* secara manual. Oleh karena itu, pada penelitian dilakukan otomatisasi manajemen konfiurasi *OpenVPN* dengan menggunakan *Ansible Playbook*. Hal ini mendorong peneliti untuk terus mengimplementasikan *Ansible Playbook* agar dapat mengotomatisasi manajemen konfigurasi *OpenVPN* dengan cara yang lebih efesien.
2) Analisa data

Dari Beberapa penelitian terdahulu dapat disimpulkan bahwa implementasi jaringan VPN dan otomatisasi konfigurasi jaringan memiliki peran penting dalam meningkatkan efisiensi, keamanan, dan kemudahan pengelolaan infrastruktur jaringan. Pada peneelitian tersebut menegaskan urgensi penggunaan teknologi yang mendukung otomatisasi dan keamanan jaringan secara terpadu untuk menjawab tantangan dalam pengelolaan jaringan modern. masih adanya kebutuhan untuk meningkatkan efisiensi, keamanan, dan kemudahan pengelolaan konfigurasi melalui otomatisasi menggunakan *tool*s seperti Ansible playbook. Pendekatan otomatisasi ini diharapkan mampu menyederhanakan proses deployment, mengurangi risiko kesalahan, serta meningkatkan kecepatan dan standarisasi penyebaran VPN di jaringan yang kompleks.



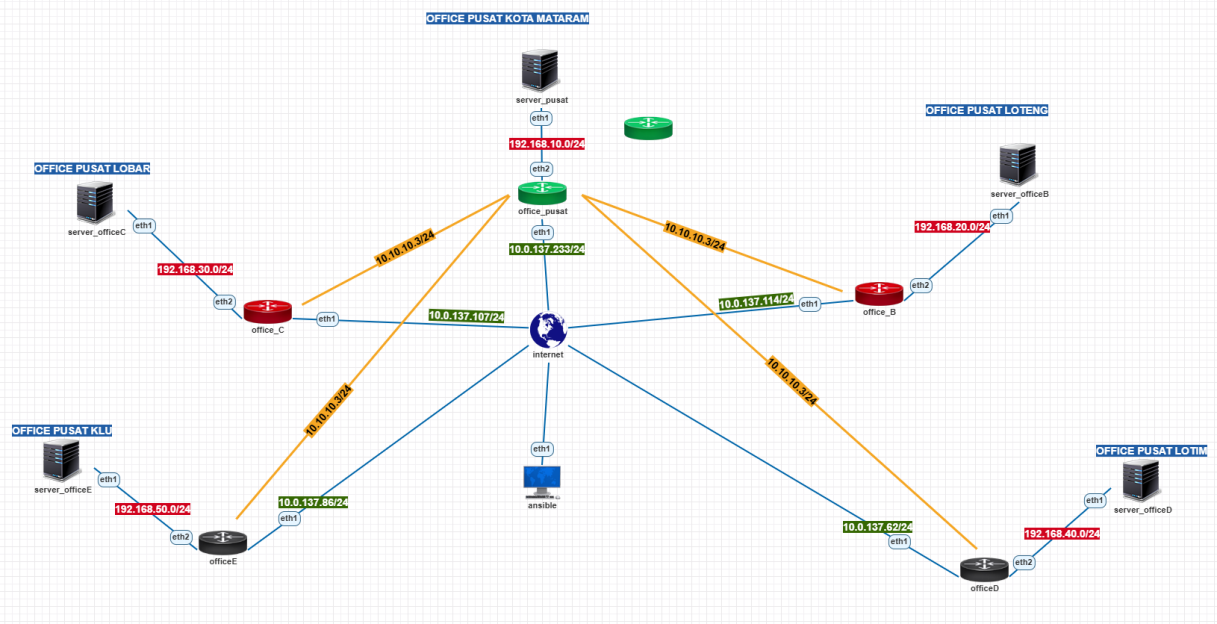
Gambar 2 kebutuhan dari sistem

B. Tahap Design

Pada tahap ini dilakukan proses perancangan sistem yang menjadi dasar dalam pelaksanaan implementasi otomatisasi konfigurasi jaringan VPN menggunakan Ansible Playbook. Desain sistem dibagi ke dalam empat bagian utama, yaitu Rancangan Jaringan Ujicoba, Rancangan Pengalamatan IP, Rancangan Sistem Otomatisasi, Kebutuhan Perangkat Keras dan Lunak. Setiap bagian dirancang untuk memastikan bahwa proses implementasi dapat berjalan dengan baik, terstruktur, dan sesuai dengan tujuan dari penelitian untuk menghasilkan struktur otomatisasi yang akan diimplementasikan.

1) Racangan jaringan ujicoba

Rancangan jaringan ujicoba yang digunakan pada penelitian ini mengadopsi model *Client*-server, di mana satu mesin bertindak sebagai *OpenVPN Server* dan beberapa mesin sebagai *Client* yang akan terhubung melalui jaringan terenkripsi. Rancangan design jaringan ujicoba yang digunakan pada penelitian ini[14][12][13], seperti terlihat pada gambar berikut.



Gambar 3 Topologi jaringa sederhana

2) Rancangan Pengalamatan IP

Pada penelitian ini diperlukan pengalamatan IP address pada setiap perangkat jaringan yang digunakan, untuk memastikan setiap perangkat dapat terhubung dan berkomunikasi dengan benar. Alamat IP jaringan yang digunakan pada penelitian ini adalah 192.168.236.194 yang diperoleh dari *server OpenVPN* secara otomatis.

Table 1 Rancangan Pengalamatan IP

NO.	Perangkat	Interface	IP Address	Netmask
1.	Pnetlab	Eth1	192.168.236.194	255.255.255.0
2.	*Router* pusat	Ether1	10.0.137.233	255.255.255.0
		Ether2	192.168.10.1	255.255.255.0
		Ovpn Tunnel	10.10.10.1	255.255.255.255
3.	Server_pusat	Ether1	192.168.10.254	255.255.255.0
4.	*Router*_officeB	Ether1	10.0.137.114	255.255.255.0

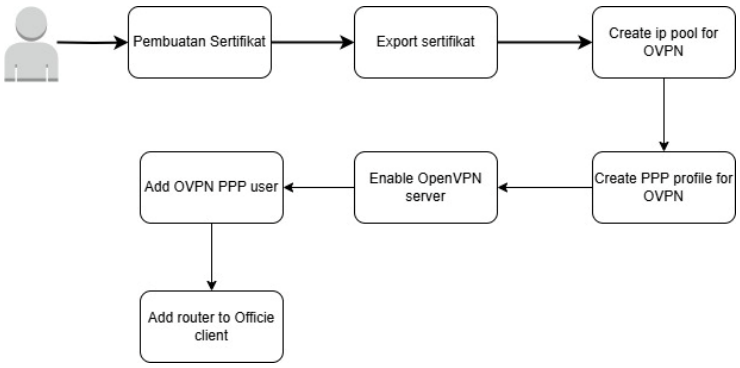
	(cabang Lobar)	Ether2	192.168.30.1	255.255.255.0
		Ovpn Tunnel	10.10.10.2	255.255.255.255
5.	Server_officeB	Ether1	192.168.30.254	255.255.255.0
6.	*Router*_officeC (cabang Loteng)	Ether1	10.0.137.107	255.255.255.0
		Ether2	192.168.20.1	255.255.255.0
		Ovpn Tunnel	10.10.10.3	255.255.255.255
7.	Server_officeC	Ether1	192.168.20.254	255.255.255.0
8.	*Router*_officeD (cabang Lotim)	Ether1	10.0.137.62	255.255.255.0
		Ether2	192.168.40.1	255.255.255.0
		Ovpn Tunnel	10.10.10.4	255.255.255.255
9.	Server_officeD	Ether1	192.168.40.254	255.255.255.0
10.	*Router*_officeE (cabang KLU)	Ether1	10.0.137.86	255.255.255.0
		Ether2	192.168.50.1	255.255.255.0
		Ovpn Tunnel	10.10.10.5	255.255.255.255
11.	Server_officeE	Ether1	192.168.50.254	255.255.255.0

3) Rancangan sistem otomatis

dirancang arsitektur otomatisasi yang meliputi pembuatan *playbook Ansible*, struktur folder dan file konfigurasi, variabel yang digunakan, serta langkah-langkah eksekusi otomatis. Perancangan ini memastikan proses konfigurasi dapat dilakukan secara otomatis dan berulang dengan konsistensi tinggi, mulai dari pembuatan sertifikat hingga konfigurasi perangkat *Mikrotik* dan verifikasi koneksi VPN.

a. Racangan sistem otomatisasi *playbook server*

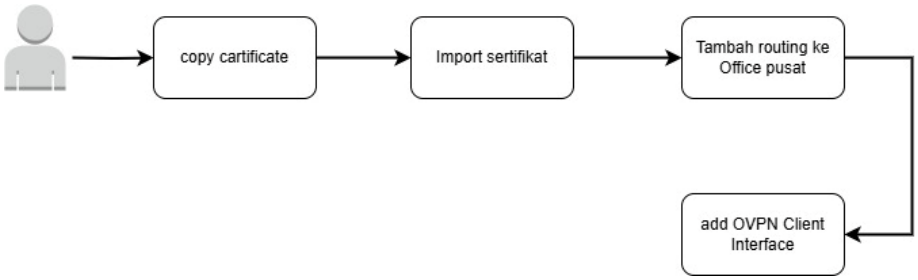
Rancangan sistem pada proses otomatisasi playbook terbagi menjadi dua sisi, yaitu sisi *client* dan *server*.



Gambar 4 Racangan alur kerja konfigurasi playbook server

a. Racangan sistem otomatisasi *playbook client*

Rancangan sistem pada proses otomatisasi playbook terbagi menjadi dua sisi, yaitu sisi *client* dan *server*.



Gambar 5 Racangan alur kerja konfigurasi playbook client

4) Kebutuhan perangkat lunak dan perangkat keras

a. Kebutuhan Perangkat Keras

- Laptop *Windows* 10
- RAM 8 GB
- Prosessor intel core i5
- Perangkat MikroTik Router (Client dan Server)
- *Network* adapter

b. Kebutuhan Perangkat Lunak

- VMware Workstation
- PnetLab
- *OpenVPN* server
- *MikroTik Router*OS

- PuTTY
- *Ansible*
- Sistem Operasi: *Linux* (Ubuntu, CentOS, Debian)

C. Tahap Simulasi Prototyping

Simulasi prototyping dilakukan sebagai bentuk implementasi awal dari sistem yang telah dirancang pada tahap sebelumnya. Tujuan dari simulasi ini adalah untuk menguji fungsionalitas dari *playbook Ansible* dalam mengotomatisasi proses instalasi, konfigurasi, dan pengujian koneksi *OpenVPN* antara *server* dan *Clien*t. Simulasi dilakukan pada enam perangkat jaringan, terdiri dari satu buah *server* yang berperan sebagai pusat layanan VPN, dan lima buah *Router* yang bertindak sebagai *Clien*t. Masing-masing *Router Clien*t dikonfigurasikan untuk terkoneksi ke *server* melalui koneksi VPN guna menguji kestabilan, keamanan, dan keandalan otomatisasi konfigurasi jaringan menggunakan *Ansible*.

1) P ersiapan Environment & Inventory

File inventory berfungsi sebagai daftar host yang akan diatur oleh Ansible, lengkap dengan IP address dan pengelompokan sesuai peran atau lokasi perangkat

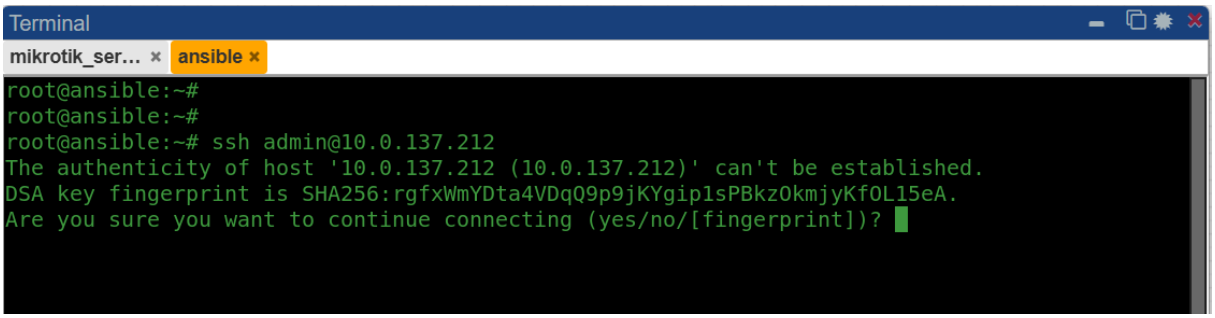
- Inventory

admin@ansible: ~

```
  GNU nano 4.8
[office_a]
10.0.137.233

[office_clients]
10.0.137.114 ansible_ssh_common_args='-o StrictHostKeyChecking=no'
10.0.137.107 ansible_ssh_common_args='-o StrictHostKeyChecking=no'
10.0.137.86 ansible_ssh_common_args='-o StrictHostKeyChecking=no'
10.0.137.62 ansible_ssh_common_args='-o StrictHostKeyChecking=no'

[office_new]
10.0.137.196 ansible_ssh_common_args='-o StrictHostKeyChecking=no'
10.0.137.23 ansible_ssh_common_args='-o StrictHostKeyChecking=no'

[all:vars]
ansible_user=admin
ansible_password="1234"
ansible_network_os=routeros
ansible_connection=network_cli
host_key_checking=false
#ansible_ssh_common_args='-o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null'
#ansible_ssh_common_args='-o StrictHostKeyChecking=no'
```

Gambar 6 File inventory

- Koneksi SSH Tanpa *Password*



Gambar 7 perintah koneksi SSH

2) Penyusunan playbook

Setelah seluruh node siap dan koneksi SSH terverifikasi, proses dilanjutkan dengan menjalankan ***playbook*** utama menggunakan perintah *Ansible*-playbook -i inventory playbook.yml. Perintah ini akan mengeksekusi seluruh role yang telah dirancang sebelumnya, dimulai dari *cert-generator* untuk membuat *Certificate* Authority (CA) dan sertifikat digital, dilanjutkan oleh *OpenVPN -server* yang mengatur konfigurasi dan instalasi *server OpenVPN* , serta *OpenVPN -Client* yang menyiapkan file konfigurasi *Client* secara otomatis.

a. Playbook server OpenVPN

admin@ansible: ~

GNU nano 4.8 playbook_server.yml

```
name: Setup OpenVPN Server (Office A)
hosts: office_a
gather_facts: no
tasks:
  - name: Create certificate templates
    community.routeros.command:
      commands:
        - /certificate add name=myCA common-name=myCA key-usage=key-cert-sign,crl-sign
        - /certificate add name=server common-name=server key-usage=digital-signature,key-encipherment,tls-server
        - /certificate add name=officeB common-name=officeB key-usage=digital-signature,key-encipherment,tls-client
        - /certificate add name=officeC common-name=officeC key-usage=digital-signature,key-encipherment,tls-client
        - /certificate add name=officeD common-name=officeD key-usage=digital-signature,key-encipherment,tls-client
        - /certificate add name=officeE common-name=officeE key-usage=digital-signature,key-encipherment,tls-client

  - name: Sign certificates
    community.routeros.command:
      commands:
        - /certificate sign myCA name=myCA
        - /certificate sign server name=server ca=myCA
        - /certificate sign officeB name=officeB ca=myCA
        - /certificate sign officeC name=officeC ca=myCA
        - /certificate sign officeD name=officeD ca=myCA
        - /certificate sign officeE name=officeE ca=myCA

  - name: Set certificates as trusted
    community.routeros.command:
      commands:
        - /certificate set myCA trusted=yes
        - /certificate set officeB trusted=yes
        - /certificate set officeC trusted=yes
        - /certificate set officeD trusted=yes
        - /certificate set officeE trusted=yes

  - name: Create ip pool for OVPN
    community.routeros.command:
      commands:
        - /ip pool add name=ovpnpool ranges=10.10.10.2-10.10.10.254

  - name: Create PPP profile for OVPN
    community.routeros.command:
      commands:
        - /ppp profile
          add local-address=10.10.10.1 name=ovpntes remote-address=ovpnpool

  - name: Enable OpenVPN server
    community.routeros.command:
```

Gambar 8 Playbook server OpenVPN

admin@ansible: ~

GNU nano 4.8 playbook

```
- name: Enable OpenVPN server
  community.routeros.command:
    commands:
      - /interface ovpn-server server
        set certificate=server cipher=blowfish128,aes256 enabled=yes \
        default-profil=ovpntes require-client-certificate=yes

- name: Add OVPN PPP user
  community.routeros.command:
    commands:
      - /ppp secret add
        name=officeB password=1234 service=ovpn profile=ovpntes
        remote-address=10.10.10.2
      - /ppp secret add
        name=officeC password=1234 service=ovpn profile=ovpntes
        remote-address=10.10.10.3
      - /ppp secret add
        name=officeD password=1234 service=ovpn profile=ovpntes
        remote-address=10.10.10.4
      - /ppp secret add
        name=officeE password=1234 service=ovpn profile=ovpntes
        remote-address=10.10.10.5

- name: Export CA and client certificates
  community.routeros.command:
    commands:
      - /certificate export-certificate myCA export-passphrase=""
      - /certificate export-certificate officeB export-passphrase="12345678"
      - /certificate export-certificate officeC export-passphrase="12345678"
      - /certificate export-certificate officeD export-passphrase="12345678"
      - /certificate export-certificate officeE export-passphrase="12345678"

- name: Add route to Office B LAN
  community.routeros.command:
    commands:
      - /ip route
        add dst-address=192.168.20.1/24 gateway=10.10.10.2
      - /ip route
        add dst-address=192.168.30.1/24 gateway=10.10.10.3
      - /ip route
        add dst-address=192.168.40.1/24 gateway=10.10.10.4
      - /ip route
        add dst-address=192.168.50.1/24 gateway=10.10.10.5
```

Gambar 9 Plabook sever OpenVPN

b. Playbook client OpeVPN

admin@ansible: ~

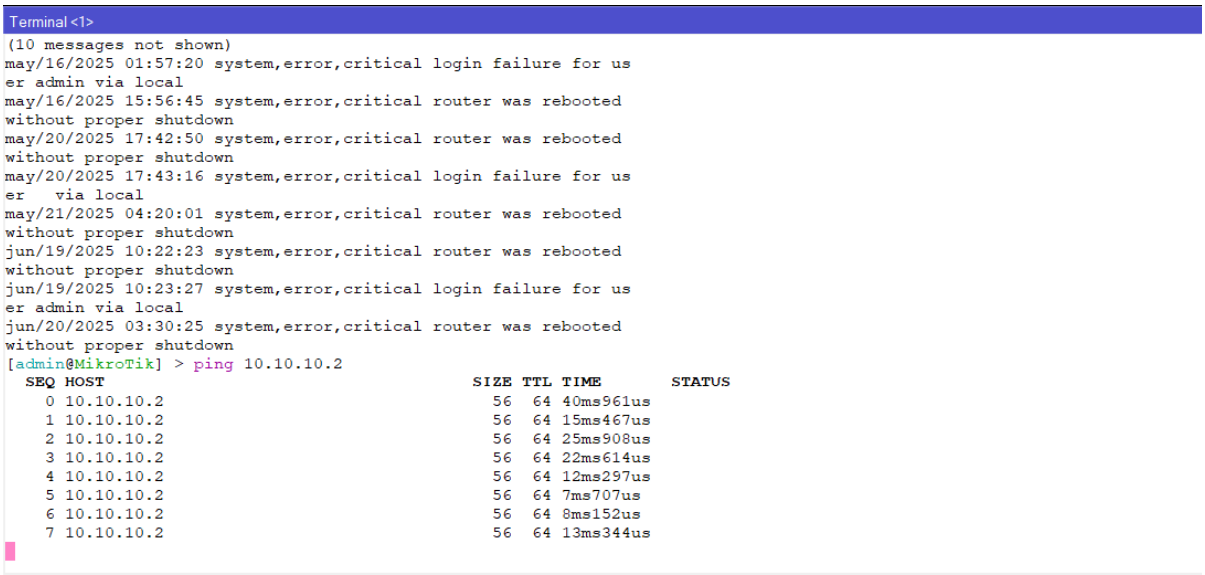
GNU nano 4.8 playbook_officeB.yml

```
- name: Setup OpenVPN Client (Office B)
  hosts: 10.0.137.114
  gather_facts: no
  tasks:
    - name: copy cartificate
      community.routeros.command:
        commands:
          - /tool fetch address=10.0.137.233 src-path=cert_export_officeB.crt user=admin mode=ftp password=1234
          - /tool fetch address=10.0.137.233 src-path=cert_export_officeB.key user=admin mode=ftp password=1234
          - /tool fetch address=10.0.137.233 src-path=cert_export_myCA.crt user=admin mode=ftp password=1234

    - name: Import sertifikat client and CA
      community.routeros.command:
        commands:
          - /certificate import name=officeB file-name=cert_export_officeB.crt passphrase="12345678"
          - /certificate import name=officeB file-name=cert_export_officeB.key passphrase="12345678"
          - /certificate import name= myCA file-name=cert_export_myCA.crt passphrase=""

    - name: Tambah routing ke Office A LAN
      community.routeros.command:
        commands:
          - /ip route
            add dst-address=192.168.10.0/24 gateway=10.10.10.1 \
            pref-src="" routing-table=main scope=30 suppress-hw-offload=no \
            target-scope=10

    - name: NAT hanya untuk interface WAN
      community.routeros.command:
        commands:
          - /ip firewall nat add chain=srcnat out-interface=ether1 action=masquerade

    - name: add OVPN Client Interface
      community.routeros.command:
        commands:
          - /interface ovpn-client
            add certificate=officeB cipher=aes256 connect-to=10.0.137.233 \
            name=officeB user=officeB password=1234 verify-server-certificate=yes
```

Gambar 10 Playbook client OpenVPN

3) Pengujian koneksi

Setelah seluruh konfigurasi selesai dan layanan *OpenVPN* berhasil dijalankan, dilakukan pengujian koneksi antara *Clien*t dan server. Pengujian dilakukan dengan cara mengaktifkan layanan *OpenVPN Clien*t, kemudian memeriksa apakah interface virtual tunnel terbentuk sebagai tanda koneksi VPN berhasil. Untuk memastikan koneksi berfungsi, digunakan perintah ping ke IP internal *server* VPN seperti 10.0.137.233. Jika *Clien*t dapat mengirim dan menerima balasan dari *server* melalui jalur VPN, maka simulasi dianggap berhasil. Seluruh proses ini dilakukan secara otomatis dengan minimal intervensi manual, menunjukkan keefektifan *Ansible* dalam mengelola layanan jaringan secara otomatis.



Gambar 11 Pengujian koneksi

3. Analisa Hasil Ujicoba

Analisis hasil uji coba menunjukkan bahwa penerapan otomatisasi konfigurasi jaringan *OpenVPN* dengan Ansible Playbook berjalan dengan baik dan sesuai dengan tujuan yang diharapkan. Dari hasil pengujian, dapat dilihat bahwa proses otomatisasi mampu meningkatkan efisiensi waktu, menjaga konsistensi konfigurasi, serta memastikan stabilitas dan keamanan koneksi VPN secara efektif

a) Perbandingan waktu pembuatan

menunjukkan hasil uji coba Pembuatan perbandingan waktu konfigurasi *OpenVPN* yang dilakukan secara manual dan otomatis menggunakan *Ansible. H*al ini dilakukan sebanyak 3 kali percobaan untuk menetukan Estimasi waktu.

Perbandingan dicatat meliputi tercepat, terlama, rata – rata waktu yang di butuhkan untuk meakukan konfigurasi.

Tabel 1 Perbandigam waktu pembuatan OpenVPN

Percobaan	*Server*		*Client* s							
			Client B		*Client* C		*Client* D		*Client* E	
	Manual	Otomatis	Manual	Otomatis	Manual	Otomatis	Manual	Otomatis	Manual	Otomatis
1	20 menit 25 Detik	6 menit 10 Detik	10 menit 30 Detik	1 menit 50 Detik	10 menit 20 Detik	1 menit 45 Detik	10 menit 12 Detik	1 menit 43 Detik	9 menit 58 Detik	1 menit 45 Detik
2	15 menit 58 Detik	5 menit 50 Detik	10 menit 40 Detik	2 menit	10 menit 21 Detik	1 menit 47 Detik	10 menit 18 Detik	1 menit 47 Detik	10 menit 07 Detik	1 menit 45 Detik
3	18 menit 15 Detik	5 menit 30 Detik	10 menit 25 Detik	2 menit	10 menit 15 Detik	1 menit 50 Detik	10 menit 11 Detik	1 menit 45 Detik	9 menit 50 Detik	1 menit 40 Detik
Waktu tercepat	15 menit 58 Detik	5 menit 30 Detik	10 menit 25 Detik	1 menit 50 detik	10 menit 15 Detik	1 menit 45 Detik	10 menit 11 Detik	1 menit 43 Detik	9 menit 50 Detik	1 menit 40 Detik
Waktuk terlama	20 menit 25 Detik	6 menit 10 Detik	10 menit 40 Detik	2 menit	10 menit 21 Detik	1 menit 50 Detik	10 menit 18 Detik	1 menit 47 Detik	10 menit 07 Detik	1 menit 45 Detik
Rata – rata	15–20 menit	5–6 menit	10 menit	1-2 menit	10 menit	1-2 menit	10 menit	1-2 menit	9-10 menit	1-2 menit

Tabel 2 Perbandingan waktu penghapusan OpenVPN

Percobaan	*Client* s							
	Client B		*Client* C		*Client* D		*Client* E	
	Manual	Otomatis	Manual	Otomatis	Manual	Otomatis	Manual	Otomatis
1	2 menit 30 detik	1 mneit 05 Detik	2 menit 20 detik	1 menit 04 Detik	2 menit 22 detik	1 menit	2 menit 18 detik	1 menit 05 Detik
2	2 menit 23 detik	58 Detik	2 menit 21 Detik	57 Detik	2 menit 23 Detik	58 Detik	2 menit 20 Detik	56 Detik
3	2 menit 20 detik	1 menit	2meni t 15 Detik	57 Detik	2 menit 21 Detik	1 menit	2 menit 25 Detik	1 menit
Waktu tercepat	2 menit 20 detik	58 Detik	2 menit 15 Detik	57 Detik	2 menit 21 Detik	58 Detik	2 menit 18 Detik	56 Detik
Waktuk terlama	2 menit 30 detik	1 mneit 05 Detik	2 menit 21 Detik	1 menit 04 Detik	2 menit 23 Detik	1 menit 47 Detik	2 menit 20 Detik	1 menit 05 Detik
Rata – rata	2 menit	50-60 detik	2 menit	50-60 detik	2 menit	50-60 detik	2 menit	50-60 detik

b) Hasil analisa pembuatan Clien baru

Penambahan *client* baru dalam sistem ini dapat dilakukan dengan mudah melalui eksekusi ulang playbook Ansible yang telah disesuaikan. Cukup dengan menambahkan alamat IP router *client* ke dalam file *inventory* dan menentukan parameter yang dIPerlukan, proses konfigurasi—mulai dari pembuatan sertifikat, pengaturan koneksi VPN, hingga penambahan route—akan dilakukan secara otomatis tanpa perlu konfigurasi manual pada perangkat tersebut.

- Hasil analisa *inventory* new *client*

Pada gambar 12 ini adalah file *inventory* ini dirancang untuk menambahkan *client* baru ke dalam sistem otomasi Ansible, dengan konfigurasi agar proses koneksi SSH ke MikroTik dapat berjalan tanpa hambatan verifikasi yang di mana

isi file *inventory* ini berisi alamat IP, user, dan password *client* baru serta modul yang digunakan.

```
[office_new]
10.0.137.196 ansible_ssh_common_args='-o StrictHostKeyChecking=no'
10.0.137.23 ansible_ssh_common_args='-o StrictHostKeyChecking=no'

[all:vars]
ansible_user=admin
ansible_password="1234"
ansible_network_os=routeros
ansible_connection=network_cli
host_key_checking=false
```

Gambar 12 Penambahan *client* baru

- Hasil analisa konfigurasi *server* untuk *client* baru

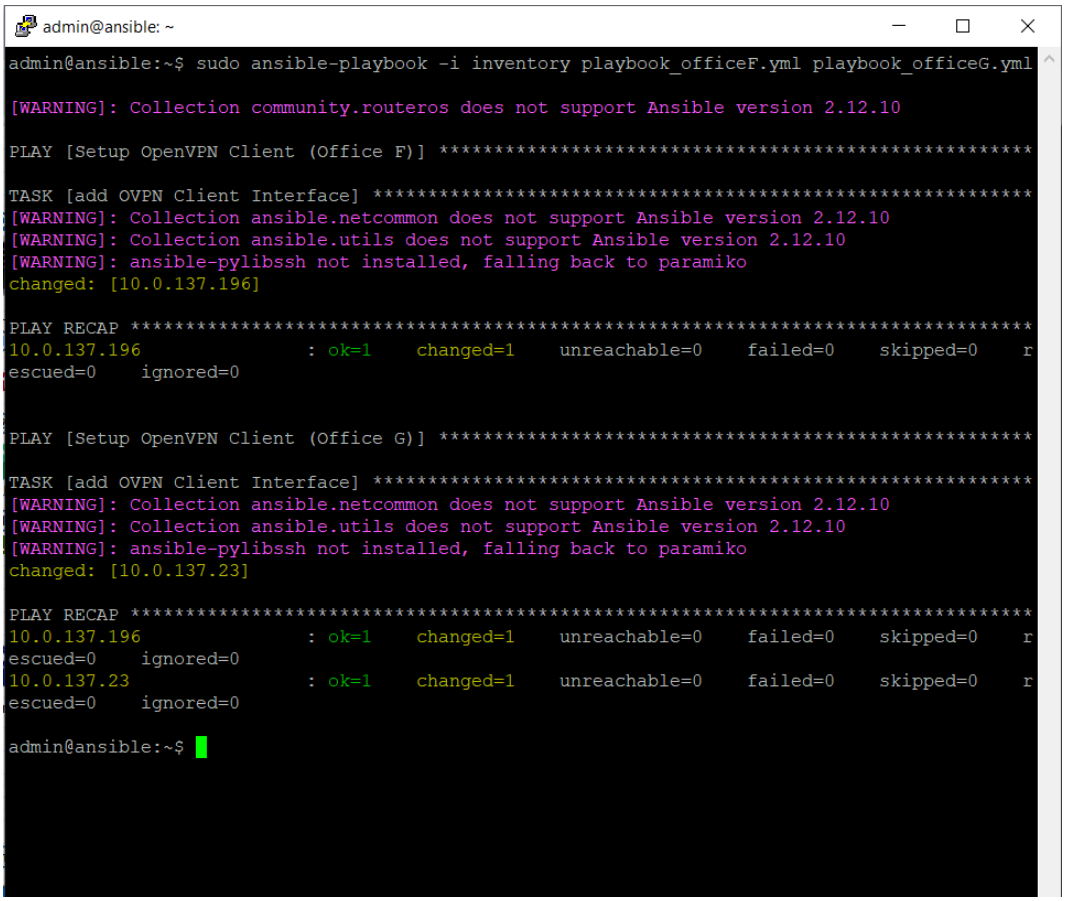
Gambar 13 menunjukkan bahwa proses penambahan *client* baru pada sisi *server OpenVPN* telah berhasil dilakukan secara otomatis menggunakan Ansible. Semua langkah penting mulai dari pembuatan dan penandatanganan sertifikat, ekspor file penting, hingga penambahan route jaringan telah dijalankan tanpa kendala. Ini membuktikan bahwa sistem otomatisasi bekerja dengan efektif, efisien, dan siap untuk menangani penambahan *client* secara cepat dan konsisten.

```
admin@ansible:~$ sudo ansible-playbook -i inventory playbook_add_client.yml
[WARNING]: Collection community.routeros does not support Ansible version 2.12.10

PLAY [add  new client(Office A)] ************************************************************

TASK [Create certificate templates] *********************************************************
[WARNING]: Collection ansible.netcommon does not support Ansible version 2.12.10
[WARNING]: Collection ansible.utils does not support Ansible version 2.12.10
[WARNING]: ansible-pylibssh not installed, falling back to paramiko
changed: [10.0.137.233]

TASK [sign client certificates] *************************************************************
[WARNING]: Collection ansible.netcommon does not support Ansible version 2.12.10
[WARNING]: Collection ansible.utils does not support Ansible version 2.12.10
changed: [10.0.137.233]

TASK [Export CA and client certificates] ****************************************************
[WARNING]: Collection ansible.netcommon does not support Ansible version 2.12.10
[WARNING]: Collection ansible.utils does not support Ansible version 2.12.10
changed: [10.0.137.233]

TASK [add new secrets] **********************************************************************
[WARNING]: Collection ansible.netcommon does not support Ansible version 2.12.10
[WARNING]: Collection ansible.utils does not support Ansible version 2.12.10
changed: [10.0.137.233]

TASK [add new ip route] *********************************************************************
[WARNING]: Collection ansible.netcommon does not support Ansible version 2.12.10
[WARNING]: Collection ansible.utils does not support Ansible version 2.12.10
changed: [10.0.137.233]

PLAY RECAP **********************************************************************************
10.0.137.233               : ok=5    changed=5    unreachable=0    failed=0    skipped=0    r
escued=0    ignored=0

admin@ansible:~$
```

Gambar 13 Hasil analisa konfigurasi server untuk client baru

- Hasil analisa konfigurasi clien untuk *client* baru

Gambar 4.53 menunjukkan hasil keberhasilan penambahan dua *client* baru (Office F dan Office G) ke dalam sistem VPN menggunakan playbook otomatis Ansible. Kedua perangkat berhasil dikonfigurasi dengan *interface OpenVPN* melalui koneksi SSH menggunakan kredensial dan parameter yang telah didefinisikan dalam *inventory*. Ini menunjukkan bahwa sistem otomatisasi telah berjalan dengan baik dan siap untuk menskalakan konfigurasi ke perangkat *client* lainnya.



Gambar 14 Hasil analisa konfigurasi client baru

c) Hasil analisa Penghapusan *client*

Gambar 15 menunjukkan bahwa penghapusan konfigurasi *OpenVPN* pada sisi *client* telah berhasil dilakukan secara otomatis menggunakan Ansible. Semua elemen penting seperti *interface* VPN, route, dan sertifikat dihapus untuk memastikan

perangkat tersebut tidak lagi terhubung ke jaringan VPN. Ini merupakan bagian penting dalam proses manajemen siklus hidup *client* VPN untuk menjaga keamanan dan kontrol jaringan.

```
admin@ansible:~$ sudo ansible-playbook -i inventory playbook_remove_client.yml
[WARNING]: Collection community.routeros does not support Ansible version
2.12.10

PLAY [Setup OpenVPN Client (remove client)] ************************************

TASK [remove interface client] *************************************************
[WARNING]: Collection ansible.netcommon does not support Ansible version
2.12.10
[WARNING]: Collection ansible.utils does not support Ansible version 2.12.10
[WARNING]: ansible-pylibssh not installed, falling back to paramiko
changed: [10.0.137.114]

TASK [remove ip route] *********************************************************
[WARNING]: Collection ansible.netcommon does not support Ansible version
2.12.10
[WARNING]: Collection ansible.utils does not support Ansible version 2.12.10
changed: [10.0.137.114]

TASK [remove certificate] ******************************************************
[WARNING]: Collection ansible.netcommon does not support Ansible version
2.12.10
[WARNING]: Collection ansible.utils does not support Ansible version 2.12.10
changed: [10.0.137.114]

PLAY RECAP *********************************************************************
10.0.137.114               : ok=3    changed=3    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
```

Gambar 15 Hasil analisa Penghapusan client

4. KESIMPULAN

Berdasarkan analisis kebutuhan fungsional dan non-fungsional dari sistem otomatisasi konfigurasi jaringan VPN menggunakan Ansible, dapat disimpulkan bahwa sistem harus mampu secara otomatis membuat dan mendistribusikan sertifikat CA, sertifikat server, dan klien, serta mengonfigurasi interface OpenVPN, IP pool, profile, secrets, firewall, dan routing secara efektif. Sistem juga harus mampu mengelola koneksi client MikroTik dengan parameter yang benar, memberikan alamat IP VPN statik, menampilkan status koneksi, serta mencatat semua perubahan dalam log audit untuk keperluan keamanan dan debugging. Selain itu, sistem harus berjalan secara handal, aman, dan efisien, mampu mengotomatisasi proses tanpa intervensi manual berlebihan, serta memudahkan integrasi dengan tools monitoring jaringan. Keseluruhan, sistem ini diharapkan dapat meningkatkan efisiensi, keamanan, dan kemudahan pengelolaan jaringan VPN secara otomatis dan stabil.

REFERENSI

[1] N. M. A. Yalestia Chandrawaty and I. P. Hariyadi, "Implementasi Ansible Playbook Untuk Mengotomatisasi Manajemen Konfigurasi VLAN Berbasis VTP Dan Layanan DHCP," *Jurnal Bumigora Information Technology (BITe)*, vol. 3, no. 2, pp. 107–122, 2021, doi: 10.30812/bite.v3i2.1577.

[2] D. Asriani Siregar *et al.*, "Pengenalan Jaringan Komputer Dasar Di Smk Negeri 1 Batang Onang," *Jurnal ADAM : Jurnal Pengabdian Masyarakat* , vol. 2, no. 2, pp. 293–303, 2023, doi: 10.37081/adam.v2i2.1443.

[3] M. D. S. Antariksa and A. Aranta, "Analisis Jaringan Komputer Local Area Network (LAN) Di Rumah Sakit UNRAM," *Jurnal Begawe Teknologi Informasi (JBegaTI)*, vol. 3, no. 2, pp. 201–212, 2022, doi: 10.29303/jbegati.v3i2.748.

[4] M. Rifki Afandi, P. Hatta, A. Efendi, K. Kunci-Otomatisasi Jaringan, L. Komputer, and P. Jaringan, "Otomatisasi Perangkat Jaringan Komputer Menggunakan Ansible Pada Laboratorium Komputer," *SMARTICS Journal*, vol. 6, no. 2, pp. 48–53, 2020, [Online]. Available: https://doi.org/10.21067/smartics.v6i2.4599

[5] A. H. Prayoga, K. Insani, and D. A. Farizal, "Implementasi Otomatisasi Backup Pada Router Mikrotik Server Jaringan Internet Universitas Majalengka," *INFOTECH journal*, vol. 10, no. 2, pp. 228–233, 2024, doi: 10.31949/infotech.v10i2.10851.

[6] M. A. A. Pratama and I. P. Hariyadi, "Otomasi Manajemen dan Pengawasan Linux Container (LCX) Pada Proxmox VE Menggunakan Ansible," *Jurnal Bumigora Information Technology (BITe)*, vol. 3, no. 1, pp. 82–95, 2021, doi: 10.30812/bite.v3i1.807.

[7] F. D. J. Prasetyo, H. Sutarjo, D. Triantoro, and D. V. S. Y. Sakti, "Otomatisasi Backup Konfigurasi PerangkatJaringan Komputer Cisco," *Journal MIND Journal | ISSN*, vol. 9, no. 1, pp. 99–112, 2024, [Online]. Available: https://ejurnal.itenas.ac.id/index.php/mindjournal/article/view/11204/3665

[8] M. Fahmi, M. Maisyaroh, I. Komarudin, S. Faizah, and I. Fadhilah, "Otomatisasi Jaringan Menggunakan Script Python Untuk Penyediaan Konfigurasi Internet Dan Manajemen Mikrotik," *Bina Insani Ict Journal*, vol. 8, no. 1, p. 53, 2021, doi: 10.51211/biict.v8i1.1517.

[9] E. N. Fadhila, E. R. Gumelar, H. R. Pratama, and G. M. Suranegara, "Otomasi Konfigurasi Routing pada Router menggunakan Ansible," *Telecommunications, Networks, Electronics, and Computer Technologies (TELNECT)*, vol. 1, no. 2, pp. 93–98, 2021, [Online]. Available: https://ejournal.upi.edu/index.php/TELNECT/article/view/40806

[10] N. Evianti, M. A. Wihandar, and A. Kurniawan, "Automation Provisioning Dev-Ops Website Server Menggunakan Ansible Dan Vagrant," *Junal Nasional Informatika*, vol. 2, no. 2, pp. 72–91, 2021.

[11] I. Sting *et al.*, "B," vol. 9, no. 5, pp. 2662–2664, 2023.

[12] T. Rahman, G. M. V. T. Mariatmojo, H. Nurdin, and H. Kuswanto, "Implementasi VPN Pada VPS Server Menggunakan OpenVPN dan Raspberry Pi," *Teknika*, vol. 11, no. 2, pp. 138–147, 2022, doi: 10.34148/teknika.v11i2.482.

[13] R. Ramesh, L. Evdokimov, D. Xue, and R. Ensafi, "VPNalyzer: Systematic Investigation of the VPN Ecosystem," *29th Annual Network and Distributed System Security Symposium, NDSS 2022*, no. April, 2022, doi: 10.14722/ndss.2022.24285.

[14] K. Ghanem, S. Ugwuanyi, J. Hansawangkit, R. McPherson, R. Khan, and J. Irvine, "Security vs Bandwidth: Performance Analysis between IPsec and OpenVPN in Smart Grid," *2022 International Symposium on Networks, Computers and Communications, ISNCC 2022*, 2022, doi: 10.1109/ISNCC55209.2022.9851717.

[15] Y. Sudarman, "Analisa Penerapan Intrusion Prevention System (Ips) Untuk Mengamankan Container Docker," 2020.