# JTIM : Jurnal Teknologi Informasi dan Multimedia

**p-ISSN : 2715-2529**
**e-ISSN : 2684-9151**
*https://journal.sekawan-org.id/index.php/jtim*

# Implementasi *Software-Defined Network* Terintegrasi *Firewall* pada *Proxmox* untuk Pengontrolan Konfigurasi Jaringan dan Pengamanan Layanan *Container*

**I Putu Hariyadi [1,*], I Made Yadi Dharma [1], Raisul Azhar [1] dan Suriyati [1]**

[1] Program Studi Ilmu Komputer, Universitas Bumigora, Indonesia.

* Korespondensi: putu.hariyadi@universitasbumigora.ac.id

**Sitasi:** Hariyadi, I. P.; Dharma, I. M. Y.; Azhar, R.; Suriyati, S. (**2025**). Implementasi Software-Defined Network Terintegrasi Firewall pada Proxmox untuk Pengontrolan Konfigurasi Jaringan dan Pengamanan Layanan Container. JTIM: Jurnal Teknologi Informasi Dan Multimedia, 7(1), 107-122. https://doi.org/10.35746/jtim.v7i1.644

Diterima: 20-11-2024
Direvisi: 31-12-2024
Disetujui: 03-01-2025

**Copyright:** © 2025 oleh para penulis. Karya ini dilisensikan di bawah Creative Commons Attribution-ShareAlike 4.0 International License. (https://creativecommons.org/licenses/by-sa/4.0/).

**Abstract:** Virtualization technology has helped companies consolidate various server roles into a single physical server, reducing hardware costs. Hypervisor is a software in virtualization that is used to manage server hardware, allowing multiple Virtual Machines (VM)/Containers (CT) to run on a single physical machine. Companies face various challenges to remain competitive in the digital era, such as the need for rapid deployment of virtual guests and virtual networks on hypervisors in development, testing, and production environments, as well as securing network services. The purpose of this study is to implement SDN on hypervisors to centrally control virtual network configurations with a simple design, reducing setup and maintenance costs and time. In addition, it also implements a firewall and Virtual Private Network (VPN) based on OpenVPN and a reverse proxy to secure the hypervisor and VM/CT so that services remain available. This study presents a new approach that integrates Software-Defined Network (SDN)-based network management with comprehensive security solutions on hypervisors. This approach combines efficiency in network management and security that have rarely been focused on simultaneously in previous studies. The research method uses the Network Development Life Cycle (NDLC). The hypervisor used is Proxmox Virtual Environment (PVE) which is installed on the Virtual Private Server (VPS) provider IDCloudHost. Based on the results of the trials that have been carried out, it can be concluded that the simple zone type SDN on PVE can be used to control network configurations centrally and more simply such as routing, Dynamic Host Configuration Protocol (DHCP), Source Network Address Translation (SNAT), hostname registration and Internet Protocol (IP) from CT to forward lookup zone on the Domain Name System (DNS) server. Activating the firewall and creating rules at the cluster and CT levels from PVE and OpenVPN can protect the infrastructure when accessed both internally and externally. While the implementation of nginx reverse proxy can secure access to HTTP/HTTPS services on CT in PVE.

**Keywords:** SDN, Firewall, Reverse Proxy, Container, Proxmox

**Abstrak:** Teknologi virtualisasi telah membantu perusahaan dalam mengkonsolidasikan berbagai peran *server* ke dalam sebuah *server* fisik sehingga mengurangi biaya perangkat keras. *Hypervisor* merupakan perangkat lunak pada virtualisasi yang digunakan untuk mengelola perangkat keras *server* sehingga memungkinkan beberapa *Virtual Machine (VM)/Container (CT)* berjalan pada satu mesin fisik. Perusahaan menghadapi berbagai tantangan agar tetap kompetitif di era digital seperti kebutuhan terkait *deployment virtual guest* dan *virtual network* pada *hypervisor* dengan cepat baik di lingkungan pengembangan, pengujian maupun produksi serta pengamanan layanan jaringan. Tujuan dari penelitian ini adalah menerapkan SDN pada *hypervisor* untuk mengontrol konfigurasi *virtual network* secara terpusat dengan desain yang sederhana sehingga mengurangi biaya dan

waktu pengaturan serta pemeliharaan. Selain itu juga menerapkan *firewall* dan *Virtual Private Network (VPN)* berbasis *OpenVPN* serta *reverse proxy* untuk mengamankan *hypervisor* dan VM/CT agar layanan tetap terjaga ketersediaannya. Penelitian ini menghadirkan pendekatan baru yang mengintegrasikan pengelolaan jaringan berbasis *Software-Defined Network (SDN)* dengan solusi keamanan yang komprehensif pada *hypervisor*. Pendekatan ini memadukan efisiensi dalam manajemen jaringan dan keamanan yang jarang dijadikan fokus secara bersamaan dalam penelitian sebelumnya. Metode penelitian menggunakan *Network Development Life Cycle (NDLC)*. *Hypervisor* yang digunakan adalah *Proxmox Virtual Environment (PVE)* yang diinstalasi pada *Virtual Private Server (VPS) provider IDCloudHost*. Berdasarkan hasil ujicoba yang telah dilakukan maka dapat disimpulkan SDN bertipe *simple zone* pada PVE dapat digunakan untuk mengontrol konfigurasi jaringan secara terpusat dan lebih sederhana seperti *routing*, *Dynamic Host Configuration Protocol (DHCP)*, *Source Network Address Translation (SNAT)*, registrasi *hostname* dan *Internet Protocol (IP)* dari CT ke *forward lookup zone* di *server Domain Name System (DNS)*. Pengaktifan *firewall* dan pembuatan *rule* di level *cluster* dan CT dari PVE serta OpenVPN dapat memproteksi infrastruktur ketika diakses baik dari internal maupun eksternal. Sedangkan penerapan *nginx reverse proxy* dapat mengamankan akses layanan HTTP/HTTPS pada CT di PVE.

**Kata kunci:** SDN, Firewall, Reverse Proxy, Container, Proxmox

## 1. Pendahuluan

Virtualisasi merupakan teknologi yang mendasari *Cloud Computing* dan memiliki kemampuan dalam mengkonsolidasikan berbagai peran *server* ke dalam sebuah *server* fisik. *Hypervisor* mengelola dan mengalokasikan sumber daya server virtualisasi secara dinamis ke setiap VM/CT. Berbagai institusi dan perusahaan telah memanfaatkan teknologi virtualisasi melalui penggunaan *hypervisor* PVE [1], [2], [3], [4], [5], [6], [7], [8] dan *VirtualBox* [4]. Usaha Mikro, Kecil dan Menengah (UMKM) memanfaatkan *hypervisor* PVE sebagai *server cloud* untuk layanan *e-commerce* yang diinstalasi pada VM dengan sistem operasi *Ubuntu* [3]. Institusi pendidikan juga telah mengembangkan *prototype* layanan VPS dan *web hosting* lingkungan kampus berbasis PVE [6]. Selain itu *cloud* akademik yang dibangun menggunakan PVE memberikan pengalaman bagi guru dan siswa terkait penggunaan virtualisasi untuk mendukung pengajaran dan penelitian serta *tool* administrasi sistem [7]. Pemanfaatan *cloud computing* pada institusi pendidikan membuat sistem pembelajaran menjadi lebih efisien dan efektif serta meningkatkan kualitas luaran pembelajaran [9]. Untuk menjaga ketersediaan layanan maka pada *hypervisor* PVE tersebut dilakukan penerapan *firewall* berbasis *IPTables* untuk membatasi akses berdasarkan protokol dan alamat IP sumber tertentu [8]. Selain itu juga dengan menerapkan arsitektur beberapa *virtual server* yang berjalan pada PVE yang dikombinasikan dengan aplikasi *reverse proxy* dan teknik *clustering* penyimpanan untuk mengoptimalkan sumber daya [5].

Institusi dan perusahaan menghadapi berbagai tantangan terkait virtualisasi agar tetap kompetitif di era digital seperti kebutuhan terkait *deployment virtual guest* dan *virtual network* pada *hypervisor* dengan cepat baik di lingkungan pengembangan, pengujian maupun produksi. Konfigurasi *virtual network* pada *hypervisor* PVE di penelitian-penelitian terdahulu masih dilakukan secara manual baik melalui antarmuka berbasis *web* maupun *Command Line Interface (CLI)* dan tanpa sarana pengontrolan serta pemantauan terpusat sehingga menjadi tidak efisien dan efektif [1], [2], [3], [4], [5], [6], [7], [8]. Termasuk kebutuhan pengamanan karena *hypervisor* dan VM/CT memiliki risiko keamanan sebagai dampak adanya celah kerentanan yang dieksploitasi. Selain itu tantangan keamanan pada *virtual networking* apabila koneksi yang aman tidak terbentuk [10].

Penelitian ini bertujuan untuk menerapkan SDN pada *hypervisor* guna mengontrol konfigurasi *virtual network* secara terpusat dengan desain yang sederhana sehingga mengurangi biaya dan waktu pengaturan serta pemeliharaan. SDN merupakan pendekatan

jaringan modern dan *agile* yang memfasilitasi manajemen jaringan secara terpusat dan *controller* yang memberikan pandangan secara global dari keseluruhan jaringan serta mengurangi biaya operasional. SDN menjadi solusi untuk mengatasi permasalahan terkait *network virtualization* dan jaringan sebagai layanan [11]. Sistem administrator dapat mengelola dan mengontrol jaringan secara terpusat dan mengalokasikan pengalamatan IP secara dinamis serta memastikan keterjangkauan jaringan dengan lebih cepat melalui SDN [12]. Selain itu juga peneliti mengintegrasikan SDN dengan *firewall* untuk meningkatkan keamanan *cluster*, *node* dan *CT* pada PVE. *Firewall* yang diterapkan merupakan gabungan dari fitur bawaan PVE dan *IPTables* sehingga dapat saling melengkapi kelebihan dan kekurangan masing-masing *tool* tersebut. Sedangkan untuk meningkatkan keamanan akses ke layanan pada CT di *hypervisor* PVE maka diterapkan *reverse proxy* berbasis *nginx*. *Reverse proxy* tidak menangani permintaan pengguna secara langsung. Namun mengirimkan permintaan tersebut ke server *backend* dan mengirimkan hasil *server backend* ke pengguna [13]. Terakhir untuk meningkatkan keamanan *remote access* ke CT di dalam *hypervisor* PVE maka dibangun VPN berbasis OpenVPN. VPN merupakan teknologi yang dapat digunakan untuk mengamankan komunikasi melalui *Internet* ketika bekerja secara *remote* [14]. OpenVPN merupakan perangkat lunak *open source* untuk membangun VPN dan menggunakan *library OpenSSL* baik untuk enkripsi dan otentikasi serta mengenkripsi trafik yang ditransmisikan melalui *tunnel Transmission Control Protocol (TCP)* atau *User Datagram Protocol (UDP)* [15]. Pendekatan baru yang dihadirkan pada penelitian ini memadukan efisiensi dalam manajemen jaringan berbasis SDN dan keamanan berbasis *firewall*, VPN serta *reverse proxy* yang jarang dijadikan fokus secara bersamaan dalam penelitian sebelumnya.

## 2. Bahan dan Metode

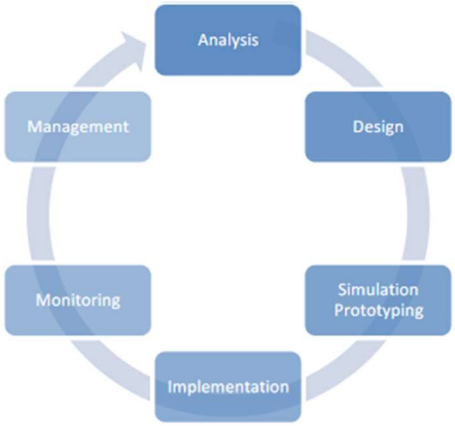
### *2.1. Bahan*

Kebutuhan perangkat keras yang diperlukan untuk mendukung penyelesaian penelitian ini adalah satu unit *laptop* dengan spesifikasi prosesor *Intel Core* i7, memori 16 GB, hardisk 512 GB dan terinstalasi sistem operasi Windows 11 serta koneksi *Internet*. Selain itu juga dibutuhkan dua *Cloud Virtual Private Server (VPS)* yang disewa di *IDCloudHost*. *Cloud* VPS tersebut difungsikan sebagai *server* DNS untuk *domain idnetbrain.com* dan diinstalasi *PVE*. *Cloud* VPS untuk server DNS memiliki spesifikasi yaitu 2 (dua) prosesor, memori 2 GB, hardisk 20 GB dan satu IP publik serta terinstalasi sistem operasi *CentOS* versi 7.6. Sedangkan *Cloud* VPS untuk server PVE memiliki spesifikasi yaitu 2 (dua) prosesor, memori 4 GB, *hardisk* 60 GB dan satu IP publik serta terinstalasi sistem operasi *Debian* versi 12. Sedangkan perangkat lunak yang dibutuhkan meliputi *VMWare Workstation* sebagai *hypervisor* pada tahap *simulation prototyping*, *putty* untuk *remote access Secure Shell (SSH)* ke VPS dan *Container (CT)*, *browser Chrome*, *CentOS* versi 7.6, PVE versi 8.2.4 sebagai *hypervisor*, OpenVPN sebagai *server VPN*, *nginx* sebagai *reverse proxy* dan *server web*, *Mikrotik Cloud Hosted Router (CHR)* sebagai *gateway Internet* dan WinSCP untuk mengunduh *profile* dari *server* OpenVPN serta CT *images Ubuntu* versi 22.04.

### *2.2. Metode*

Metode pengembangan sistem yang digunakan pada penelitian ini adalah *Network Development Life Cycle (NDLC)*. NDLC merupakan metode untuk mengembangkan jaringan komputer yang bergantung pada proses pembangunan sebelumnya seperti perencanaan strategi bisnis, siklus hidup pengembangan aplikasi dan analisis pendistribusian data. Terdapat 6 (enam) tahapan pada NDLC yaitu *analysis*, *design*, *simulation prototyping*, *implementation*, *monitoring* dan *management*, seperti terlihat pada gambar 1 [16]. Keseluruhan tahapan dari NDLC tersebut digunakan pada penelitian ini. Pada tahap *analysis* dilakukan pengumpulan data berupa artikel ilmiah dan analisa terhadap data tersebut sehingga ditemukan celah penelitian dan solusi untuk mengatasi kesenjangan terse-

but. Selanjutnya tahap *design* dilakukan pembuatan rancangan jaringan ujicoba, pengalamatan IP dan sistem sesuai hasil identifikasi di tahap *analysis*. Rancangan yang dihasilkan pada tahap *design* diuji dan divalidasi pada tahap *Simulation Prototyping* menggunakan *hypervisor VMWare Workstation*. Sedangkan pada tahap *implementation* dilakukan penerapan infrastruktur jaringan ke lingkungan produksi menggunakan VPS yang disewa di *provider IDCloudHost*. Sebaliknya pada tahap *monitoring* dilakukan pemantauan jaringan untuk mengetahui utilisasi dan mendeteksi potensi permasalahan di jaringan.




**Gambar 1.** Tahapan pada NDLC [8]

Terakhir pada tahap *management* dilakukan manajemen keamanan melalui pemanfaatan *firewall* dan VPN agar akses terhadap layanan jaringan terkontrol serta tetap terjaga ketersediaannya.

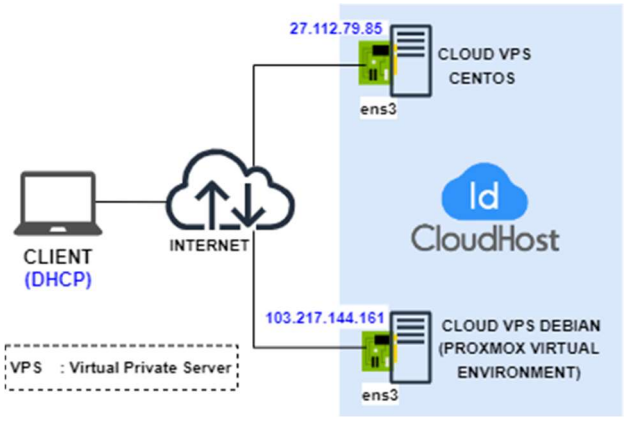
2.2.1. Tahap *Analysis*

Tahap ini terdiri dari 2 (dua) bagian yaitu pengumpulan data dan analisa data. Pengumpulan data meliputi artikel ilmiah terkait pengontrolan konfigurasi jaringan berbasis SDN dan pengamanan virtualisasi serta pengamanan akses terhadap layanan teknologi virtualisasi khususnya *CT*. Berdasarkan analisa terhadap data yang dikumpulkan maka dapat diperoleh informasi bahwa pengontrolan konfigurasi jaringan pada *hypervisor* PVE di penelitian terdahulu masih dilakukan secara manual dan tidak memiliki antarmuka terpusat untuk pengelolaan dan pemantauannya. Selain itu pengamanan yang ada hanya menggunakan *IPTables* yang diterapkan pada lingkup *cluster* dari PVE.

Mendorong ketertarikan peneliti untuk mengimplementasikan SDN sebagai pengontrolan konfigurasi jaringan khususnya *Simple Zone* pada PVE yang diintegrasikan dengan *firewall* untuk pengamanan *node* PVE dan CT. Selain itu penerapan *reverse proxy* untuk mengamankan akses layanan jaringan yang disediakan oleh *CT* pada PVE. Termasuk penerapan *OpenVPN* sebagai mekanisme untuk menjembatani kebutuhan *remote access* ke CT pada PVE melalui saluran komunikasi yang aman.

2.2.2. Tahap *Design*

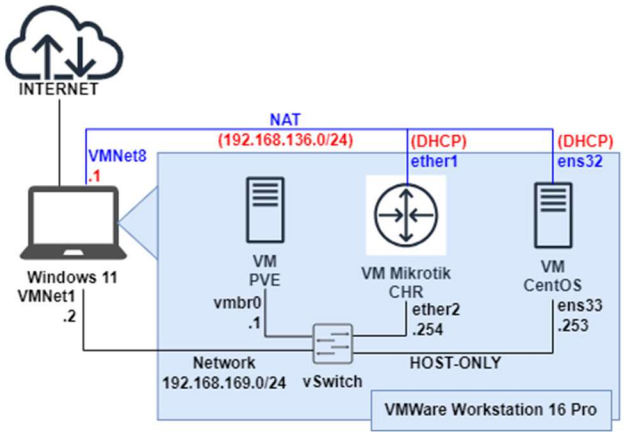
Tahap ini dilakukan pembuatan rancangan untuk memenuhi kebutuhan yang telah diidentifikasi pada tahap *analysis*. Terdapat 4 (empat) rancangan meliputi rancangan jaringan ujicoba, rancangan jaringan simulasi, rancangan pengalamatan IP dan rancangan sistem SDN pada PVE. Rancangan jaringan ujicoba untuk mengimplementasikan fitur SDN terintegrasi *firewall* dan *reverse proxy*, seperti terlihat pada gambar 2.




**Gambar 2.** Rancangan Jaringan Ujicoba

Terlihat pada rancangan jaringan ujicoba menggunakan dua *Cloud* VPS dari *IDCloudHost* yang telah terkoneksi ke *Internet* dan memiliki masing-masing satu IP Publik. *Cloud* VPS dengan IP Publik 27.113.79.85 difungsikan sebagai *server* DNS untuk *domain idnetbrain.com*. Sedangkan *Cloud* VPS dengan IP Publik 103.217.144.161 diinstalasi PVE agar berfungsi sebagai *hypervisor*. Selain itu terdapat satu *laptop* sebagai *client* untuk melakukan instalasi dan konfigurasi serta ujicoba pengaksesan layanan *Internet* yang disedikan oleh dua *Cloud* VPS.

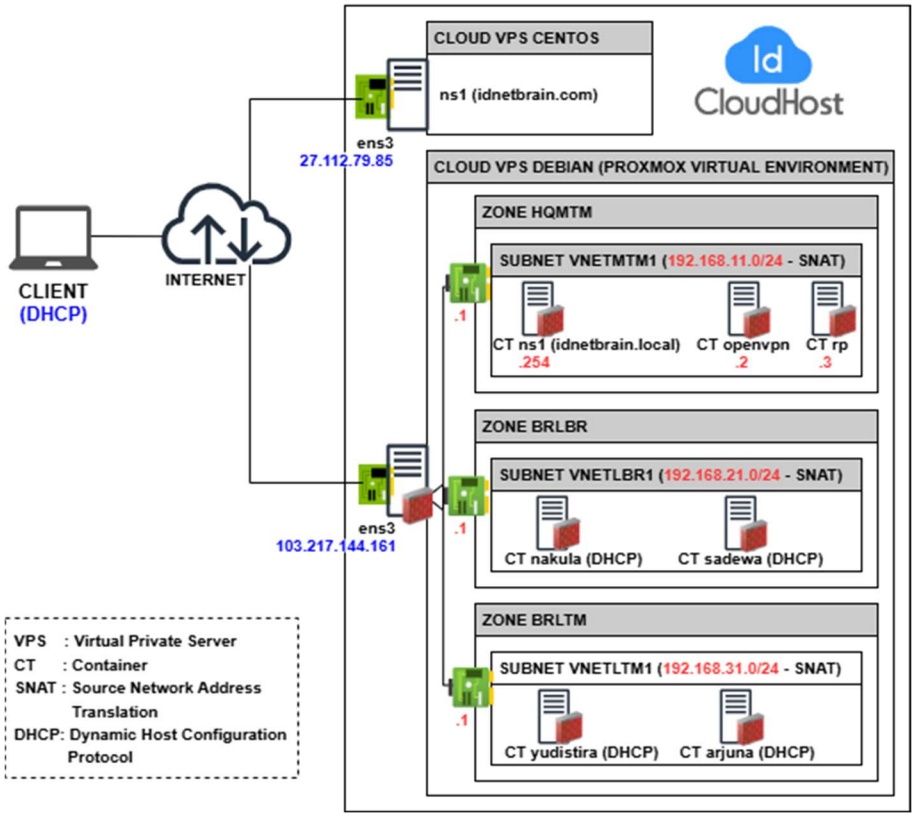
Rancangan jaringan ujicoba tersebut disimulasikan secara virtual menggunakan *VMWare Workstation Pro 16* yang diinstalasi pada sebuah *laptop* dengan sistem operasi *Windows* 11 dan terkoneksi *Internet*, seperti terlihat pada gambar 3.




**Gambar 3.** Rancangan Jaringan Simulasi

Terlihat pada *VMWare Workstation* tersebut terdapat tiga *Virtual Machine (VM)*. VM pertama diiinstalasi PVE versi 8.2.4 sebagai *hypervisor* dan memiliki satu *network adapter* bertipe *host-only* (*vmbr0*) dengan alamat IP 192.168.169.1/24. Sedangkan VM kedua diinstalasi *CentOS* sebagai *server* DNS untuk domain *idnetbrain.com* dan memiliki dua *network adapter* yaitu *ether1 (ens32)* bertipe *Network Address Translation (NAT)* dengan alokasi IP secara DHCP dan *ether2* bertipe *host-only (ens33)* dengan alamat IP 192.168.169.253/24. VM ketiga diinstalasi *Mikrotik CHR* sebagai *gateway* bagi PVE agar dapat terkoneksi ke *Internet*. VM *Mikrotik* ini memiliki dua *network adapter* yaitu *ether1* bertipe *Network Address Translation (NAT)* dengan alokasi IP secara DHCP dan *ether2* bertipe *host-only* dengan alamat IP 192.168.169.254/24. *Network connection* bertipe *host-only* menggunakan alamat *network* 192.168.169.0/24. Sedangkan yang bertipe NAT menggunakan alamat *network* 192.168.136.0/24.

Rancangan sistem SDN terintegrasi dengan *firewall* pada PVE, seperti terlihat pada gambar 4. Terlihat terdapat 3 (tiga) *zone* bertipe *simple* yang dibuat pada SDN dari PVE yaitu bernama HQMTM tanpa mengaktifkan fitur *Automatic DHCP* dan *DNS server*, BRLBR dan BRLTM dengan fitur *Automatic DHCP* yang diaktifkan dan DNS *Server*. *Zone simple* tersebut digunakan untuk mendefinisikan jaringan virtual secara terpisah. *Prefix* HQ pada nama *zone* merupakan singkatan dari *Headquarter* atau kantor pusat. Sedangkan *prefix* BR merupakan singkatan dari *Branch* atau kantor cabang. Fitur *Automatic DHCP* berfungsi untuk mengalokasikan alamat IP secara dinamis ke CT yang terhubung ke *Virtual Network (VNet)* dalam *zone* tersebut yang pengelolaannya melalui fitur *IP Address Management (IPAM)* dari SDN. Sebaliknya pengaktifan fitur *DNS Server* berfungsi untuk meregistrasi secara otomatis *hostname* dan alamat IP DHCP dari setiap CT ke *server* DNS dengan *domain idnetbrain.local*. Pada *zone* HQMTM dibuat *VNet* bernama VNETMTM1 dan di *VNet* tersebut dibuat *subnet* dengan alamat *network* 192.168.11.0/24. Terdapat tiga CT yang terhubung ke VNETMTM1 tersebut yaitu CT *ns1.idnetbrain.local* dan *openvpn.idnetbrain.local* serta *rp.idnetbrain.local*.




**Gambar 4.** Rancangan Sistem SDN terintegrasi Firewall pada PVE

Sedangkan pada *zone* BRLBR dibuat *VNet* bernama VNETLBR1 dan di *VNet* tersebut dibuat *subnet* dengan alamat *network* 192.168.21.0/24 serta rentang alamat IP yang didistribusikan melalui DHCP yaitu 192.168.21.100-192.168.21.200. Terdapat dua CT yang terhubung ke VNETLBR1 tersebut yaitu CT *nakula* dan *sadewa*. Sebaliknya pada *zone* BRLTM dibuat *VNet* bernama VNETLTM1 dan di *VNet* tersebut dibuat *subnet* dengan alamat *network* 192.168.31.0/24 serta rentang alamat IP yang didistribusikan melalui DHCP yaitu 192.168.31.100-192.168.31.200. Terdapat dua CT yang terhubung ke VNETLTM1 tersebut yaitu CT *yudistira* dan *arjuna*. Selain itu pada setiap *VNet* dari masing-masing *zone* mengaktifkan *Source Network Address Translation (SNAT)* agar CT yang terhubung ke *VNet* tersebut dapat terkoneksi ke *Internet*.

Rancangan pengalamatan IP pada SDN PVE menggunakan 3 (tiga) alamat *network Class* C yaitu 192.168.11.0/24 untuk *zone* HQMTM, 192.168.21.0/24 untuk *zone* BRLBR dan 192.168.31.0/24 untuk *zone* BRLTM. Alamat IP pertama dari setiap *network* tersebut digunakan oleh *interface* VNet masing-masing *zone* dan difungsikan sebagai *gateway* bagi CT. Pengalamatan IP untuk setiap CT pada masing-masing *zone* dialokasikan secara dinamis menggunakan DHCP. Sedangkan setiap *Cloud* VPS memiliki dua alamat yaitu IP Publik dan IP *Private* dengan alamat *network* 10.8.27.0/24. Tabel 1 memperlihatkan detail alokasi pengalamatan IP pada setiap perangkat yang diimplementasikan secara riil pada dua *Cloud* VPS, *Client Internet* dan CT di PVE.

**Tabel 1.** Pengalamatan IP pada Setiap Perangkat

| Perangkat | Interface | Alamat IP/Subnet-mask | Gateway |
|---|---|---|---|
| VPS CentOS | eth0 | 10.8.27.227/24 | 10.8.27.227 |
| | eth0 | 27.112.79.85 | |
| VPS Debian (PVE) | eth0 | 10.8.27.54/24 | 10.8.27.1 |
| | eth0 | 103.217.144.161 | |
| Client Internet | Ethernet | DHCP | |
| CT ns1 | eth0 | 192.168.11.254/24 | 192.168.11.1 |
| CT openvpn | eth0 | 192.168.11.2/24 | 192.168.11.1 |
| CT rp | eth0 | 192.168.11.3/24 | 192.168.11.1 |
| CT nakula | eth0 | DHCP | |
| CT sadewa | eth0 | DHCP | |
| CT yudistira | eth0 | DHCP | |
| CT arjuna | eth0 | DHCP | |

2.2.3. Tahap *Simulation Prototyping*

Pada tahap ini dilakukan pengujian dan validasi terhadap rancangan sistem SDN di lingkungan simulasi menggunakan *VMWare Workstation*. Selain itu juga melakukan *prototyping* dengan membangun versi kecil dari sistem SDN untuk menguji konfigurasi.

2.2.4. Tahap *Implementation*

Pada tahap ini dilakukan penerapan infrastruktur jaringan ke lingkungan produksi melalui konfigurasi pada setiap VPS berdasarkan rancangan pengalamatan IP dan rancangan sistem SDN yang telah disimulasikan pada tahap sebelumnya. Selain itu juga dilakukan pengujian fungsional dan *troubleshooting* sehingga layanan dapat diakses oleh pengguna yaitu *Client Internet*. Parameter yang digunakan untuk mengukur keberhasilan implementasi adalah fitur IPAM pada SDN memperlihatkan alokasi pengalamatan IP secara dinamis ke setiap CT pada PVE secara terpusat dan pada *server* DNS terbuat secara otomatis entri *Resource Record (RR)* IN A pada *forward lookup zone* dari domain yang digunakan. Selain itu ketika *firewall* diaktifkan maka hanya layanan dengan *rule* yang secara eksplisit dibuat pada *firewall* yang dapat diakses layanannya oleh *client Internet*.

2.2.5. Tahap *Monitoring*

Pada tahap ini dilakukan pemantauan jaringan secara terus-menerus untuk memastikan jaringan beroperasi seperti yang diharapkan terutama terkait fitur SDN pada PVE seperti *IP Address Management (IPAM)* dan utilitasi sumber daya di PVE. Selain itu juga untuk mendeteksi lebih awal terkait potensi permasalahan di jaringan.

2.2.6. Tahap Management

Pada tahap ini dilakukan manajemen keamanan agar jaringan tetap dapat beroperasi melalui pengelolaan *firewall* pada PVE. Selain itu juga dilakukan manajemen kebijakan

terkait pengontrolan akses melalui pengelolaan VPN menggunakan *OpenVPN* dan *reverse proxy* menggunakan *nginx*.

## 3. Hasil

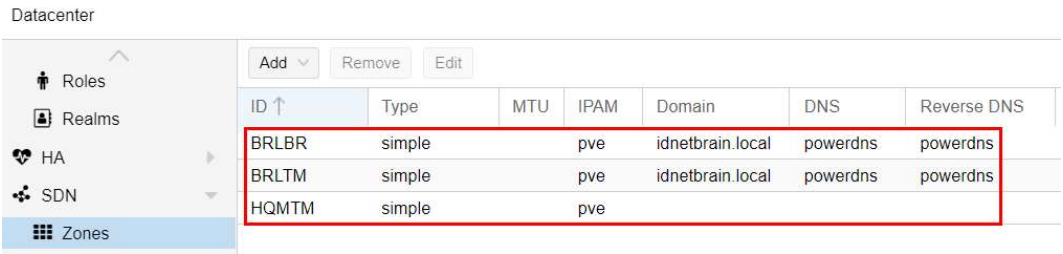
Pada bagian ini membahas tentang hasil dari instalasi dan konfigurasi serta verifikasi terhadap konfigurasi yang telah dilakukan pada VPS *CentOS*, VPS *Debian* dan CT pada PVE serta *Client Internet*.

### *3.1. Hasil Instalasi dan Konfigurasi DNS pada VPS CentOS*

VPS *CentOS* dikonfigurasi sebagai server DNS untuk *domain idnetbrain.com* menggunakan *Berkeley Internet Name Domain (BIND)* versi 9.11.4. BIND memiliki *file* konfigurasi utama bernama *named.conf* yang terdapat di direktori */etc*. Pada *file named.conf* tersebut dideklarasikan dua *zone* yaitu *forward lookup zone* untuk memetakan nama domain *idnetbrain.com* ke alamat IP dan *reverse lookup zone* untuk memetakan alamat IP ke nama domain. Kedua *zone* tersebut bertipe *master* sehingga bertindak sebagai *primary name server* untuk domain *idnetbrain.com*. Detail konfigurasi *forward lookup zone* terdapat pada *file /var/named/idnetbrain.com.zone* yang memuat 4 (empat) entri *Resource Record (RR) IN A* yang memetakan subdomain dari domain *idnetbrain.com* yaitu nakula, sadewa, arjuna dan yudistira ke alamat IP publik 103.217.144.161.

### *3.2. Hasil Instalasi dan Konfigurasi PVE pada VPS Debian*

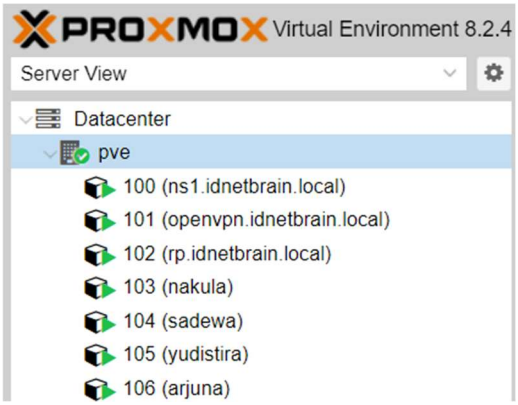
PVE yang berhasil diinstalasi pada VPS *Debian* adalah versi 8.2.4. Pada PVE tersebut dilakukan instalasi dan konfigurasi SDN, pembuatan *Simple Zone*, pembuatan CT, konfigurasi DNAT pada *IPTables* agar layanan VPN dan *web* pada CT dengan IP *Private* dapat diakses menggunakan IP Publik dari PVE. Hasil dari pembuatan *Simple Zone* berdasarkan rancangan sistem SDN, seperti terlihat pada gambar 5.




**Gambar 5.** Simple Zone pada SDN PVE

Terlihat terdapat 3 (tiga) *zone* bertipe *simple* yaitu BRLBR, BRLTM dan HQMTM. Ketiga *zone* tersebut menggunakan *IP Address Management (IPAM)* untuk mengelola IP pada *zone* tersebut. Khusus untuk *zone* BRLBR dan BRLTM juga dilakukan pengaturan *options* pada *zone* tersebut agar ketika terdapat CT yang ditambahkan pada *zone* tersebut maka akan secara otomatis meregistrasi *hostname* dan alamat IP ke *forward lookup zone* dari *server* DNS *idnetbrain.local*. Sebaliknya ketika CT dihapus maka secara otomatis menghapus registrasi *hostname* dan alamat IP dari *forward lookup zone* pada *server* DNS *idnetbrain.local*. *Options* yang diatur meliputi *domain* menggunakan *idnetbrain.local*, DNS dan *Reverse* DNS menggunakan "*powerdns*".

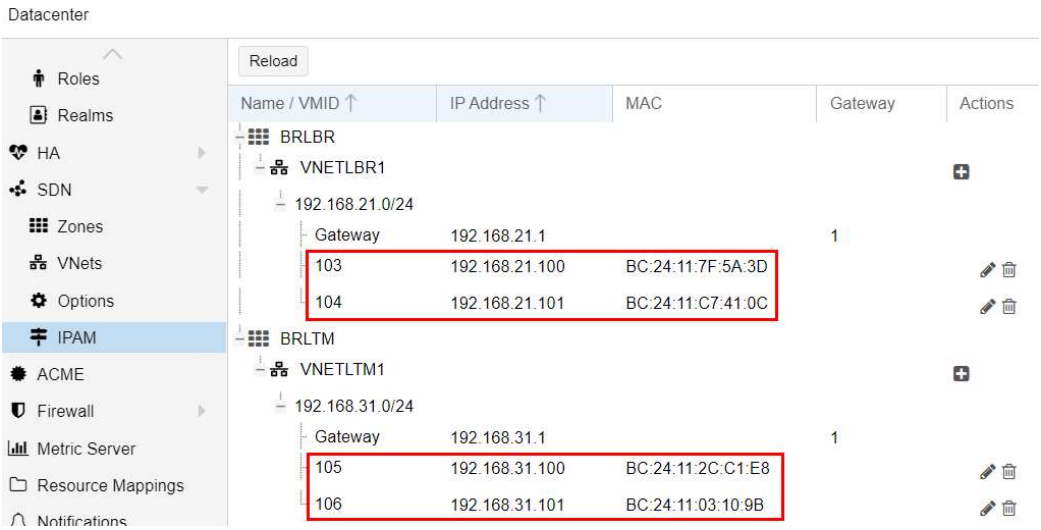
Hasil pembuatan dan pengaktifan CT pada PVE sehingga layanan didalamnya dapat diakses, seperti terlihat pada gambar 6.




**Gambar 6.** CT pada PVE.

Terlihat terdapat 7 (tujuh) CT yaitu CT ID 100 dengan *hostname ns1.idnetbrain.local* yang difungsikan sebagai *server* DNS untuk *domain idnetbrain.local*, CT ID 101 dengan *hostname openvpn.idnetbrain.local* yang difungsikan sebagai *server OpenVPN*, CT ID 102 dengan *hostname rp.idnetbrain.local* yang difungsikan sebagai *reverse proxy*, CT ID 103 dengan *hostname nakula* dan CT ID 104 dengan *hostname sadewa* berfungsi sebagai *server web* yang dihubungkan ke *zone* BRLTM, CT ID 105 dengan *hostname yudistira* dan CT ID 106 dengan *hostname arjuna* berfungsi sebagai *server web* dan dihubungkan ke *zone* BRLTM.

Hasil pembuatan VNet pada setiap *simple zone* dan alokasi pengalamatan IP secara dinamis ke setiap CT dapat diamati melalui menu IPAM dari SDN PVE, seperti terlihat pada gambar 7. Terlihat CT 103 (nakula) memperoleh alamat IP 192.168.21.100. Sedangkan CT 104 (sadewa) memperoleh alamat IP 192.168.21.101. Kedua CT tersebut terhubung ke VNETLBR1 dari *zone* BRLBR dan menggunakan *gateway* 192.168.21.1.



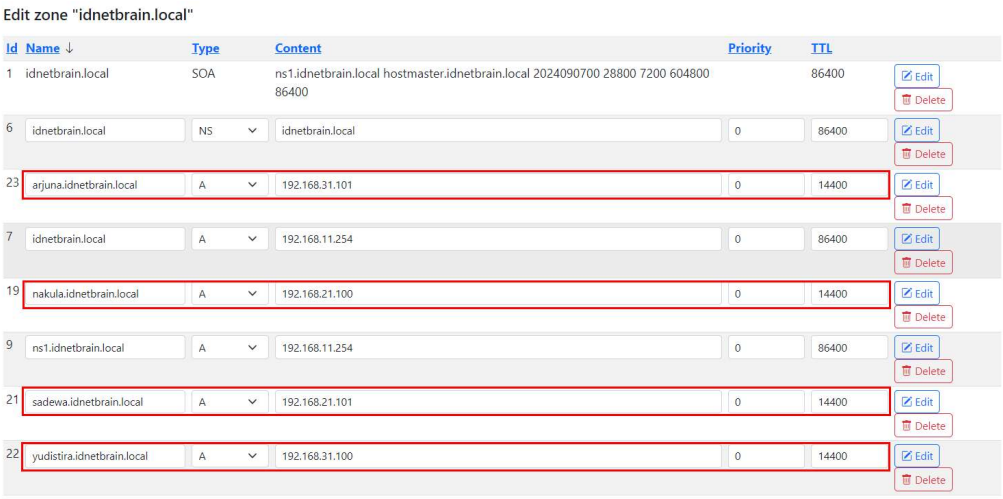

**Gambar 7.** IPAM dari SDN.

Selain itu terlihat juga CT 105 (yudistira) memperoleh alamat IP 192.168.31.100. Sedangkan CT 106 (arjuna) memperoleh alamat IP 192.168.31.101. Kedua CT tersebut terhubung ke VNETLTM1 dari *zone* BRLTM dan menggunakan *gateway* 192.168.31.1.

### *3.3. Hasil Instalasi dan Konfigurasi CT ns1 pada PVE*

CT *ns1* dengan ID 100 difungsikan sebagai *Name Server* yang dibangun menggunakan *package powerdns* dan *poweradmin* sebagai antarmuka manajemen berbasis *web*. Terdapat empat *zone* yang dibuat pada *powerdns* yaitu satu *forward lookup zone* bernama *idnetbrain.local* dan tiga *reverse lookup zone* bernama *11.168.192.in-addr.arpa*,

*21.168.192.in-addr.arpa, 31.168.192.in-addr.arpa*. Detail konten *forward lookup zone* dengan nama *idnetbrain.local* setelah dilakukan penambahan CT di PVE, seperti terlihat pada gambar 8.




**Gambar 8.** *Forward Lookup Zone idnetbrain.local* pada *PowerDNS*

Terlihat terdapat 4 (empat) entri *Resource Record (RR) IN A* yang ditandai dengan kotak berwarna merah yaitu subdomain *arjuna.idnetbrain.com* dengan alamat IP 192.168.31.101, subdomain *nakula.idnetbrain.com* dengan alamat IP 192.168.21.100, subdomain *sadewa.idnetbrain.com* dengan alamat IP 192.168.21.101 dan subdomain *yudistira.idnetbrain.com* dengan alamat IP 192.168.31.100. Entri tersebut terbuat secara otomatis pada *powerdns* ketika CT tersebut dibuat di PVE.

### *3.4. Hasil Instalasi dan Konfigurasi CT OpenVPN pada PVE*

Instalasi dan konfigurasi *server OpenVPN* pada CT 101 dilakukan menggunakan *script openvpn-install* yang dieksekusi dengan perintah *bash openvpn-install.sh*. Terdapat beberapa data yang diperlukan ketika proses instalasi dan konfigurasi *server OpenVPN* tersebut meliputi alamat IP Publik menggunakan 103.217.144.161, protokol yang digunakan oleh *OpenVPN* yaitu UDP, nomor *port* yang digunakan oleh *OpenVPN* yaitu 1194 dan pilihan *server* DNS untuk *client* yaitu 1.1.1.1 serta nama *profile* untuk *client OpenVPN* pertama yaitu *research*. Keberhasilan instalasi dan konfigurasi *server OpenVPN* dapat diverifikasi dengan mengeksekusi perintah *bash openvpn-install.sh*. Tampil pesan *OpenVPN is already installed* yang menginformasikan bahwa *OpenVPN* telah terinstalasi pada CT 101. Konfigurasi dari *server* OpenVPN tersebut tersimpan di *file /etc/openvpn/server/server.conf*. Selain itu status *service* dari *OpenVPN* yang telah aktif atau berjalan dapat diverifikasi dengan mengeksekusi perintah *systemctl status openvpn* sehingga memperlihatkan pesan *active (running)*. *File profile* pertama untuk *client OpenVPN* yang dibuat saat instalasi dan konfigurasi *OpenVPN* terdapat di */root/research.ovpn*. *File* tersebut diperlukan oleh *Client Internet* sebagai *client OpenVPN* sehingga dapat terkoneksi ke *server OpenVPN*.

### *3.5. Hasil Instalasi dan Konfigurasi CT Reverse Proxy pada PVE*

Terdapat 3 (tiga) langkah instalasi dan konfigurasi yang dilakukan pada CT *Reverse Proxy* (rp) yaitu menginstalasi *package nginx*, membuat *file* konfigurasi *virtual host* dan menginstalasi sertifikat SSL *Let's Encrypt* untuk 4 (empat) subdomain meliputi *nakula.idnetbrain.com, sadewa.idnetbrain.com, yudisitira.idnetbrain.com, arjuna.idnetbrain.com*. Versi *nginx* yang berhasil diinstalasi pada CT *Ubuntu* 22.04 adalah 1.18.0. File konfigurasi dari *virtual host* terdapat pada direktori */etc/nginx/sites-available* dan memiliki *symbolic link* ke

direktori */etc/nginx/sites-enabled* untuk mengaktifkan konfigurasi tersebut. Salah satu contoh hasil pembuatan *file virtual host* dan instalasi SSL *Let's Encrypt* untuk subdomain *nakula.idnetbrain.com* bernama *nakula.idnetbrain.com.conf*, seperti terlihat pada gambar 9.

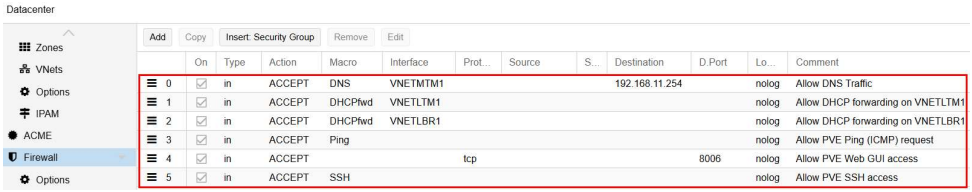
```
root@rp:/etc/nginx/sites-available# cat nakula.idnetbrain.com.conf
server {
    server_name nakula.idnetbrain.com;

    location / {
        proxy_pass http://nakula.idnetbrain.local;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto https;
    }

    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/nakula.idnetbrain.com/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/nakula.idnetbrain.com/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}
```

**Gambar 9.** Konten *File Virtual Host nakula.idnetbrain.com*

Terlihat pada *file* konfigurasi *virtual host* tersebut, *reverse proxy* diaktifkan melalui *directive proxy_pass* dan dilakukan penyesuaian nilai dari *field header* menggunakan *directive proxy_set_header* dari *nginx*. Selain itu juga terlihat *file* sertifikat SSL *Let's Encrypt* untuk subdomain *nakula.idnetbrain.com* tersimpan pada direktori */etc/letsencrypt/live/nakula.idnetbrain.com* masing-masing bernama *fullchain.pem* dan *privkey.pem*. Ketika terdapat permintaan HTTP/HTTPS ke subdomain *nakula.idnetbrain.com* maka permintaan tersebut akan dikirimkan ke CT *nakula* dengan alamat *nakula.idnetbrain.local* di dalam PVE. *Client Internet* tidak berkomunikasi langsung dengan CT *nakula* namun melalui perantara yaitu CT rp yang bertindak sebagai *reverse proxy*.

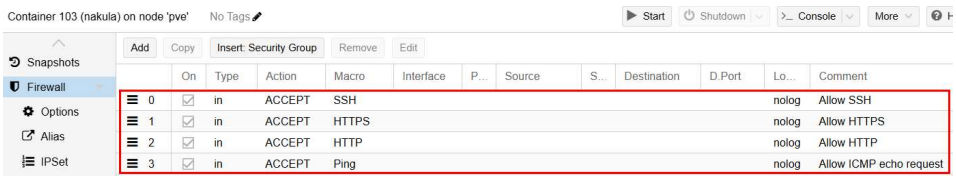
### 3.6. *Hasil Instalasi dan Konfigurasi Firewall*

*Firewall* dikonfigurasi pada lingkup *cluster* dan CT menggunakan fitur *firewall* bawaan dari PVE. Sedangkan DNAT dikonfigurasi menggunakan *IPTables* karena *firewall* PVE belum mendukung pengaturan tersebut. Hasil konfigurasi *firewall rule* pada lingkup *cluster* dari PVE, seperti terlihat pada gambar 10.




**Gambar 10.** Konfigurasi *Firewall* pada PVE *Cluster*

Terlihat terdapat 6 (enam) *firewall rule* yang dibuat pada *cluster* dari PVE yaitu *rule* untuk mengijinkan akses DNS pada *interface* VNETMTM1 dengan tujuan IP dari *server* DNS internal yaitu 192.168.11.254, DHCP pada *interface* VNETLTM1 dan VNETLBR1, ICMP *Ping* dan SSH menggunakan fitur *macro* dari *firewall* PVE serta *WebGUI* dari PVE menggunakan *transport* TCP pada *port* 8006. Sedangkan hasil konfigurasi *firewall rule* pada lingkup CT pada PVE dengan contoh di CT nakula, seperti terlihat pada gambar 11.




**Gambar 11.** Konfigurasi *Firewall* pada CT nakula

Terlihat terdapat 4 (empat) *firewall rule* yang dibuat pada CT *nakula* yaitu *rule* untuk mengijinkan akses SSH, HTTPS, HTTP dan ICMP *Ping* menggunakan fitur macro dari *firewall* PVE. Terakhir cuplikan hasil konfigurasi *IPTables* terkait DNAT pada *node* PVE, seperti terlihat pada gambar 12.

```
root@pve:~# iptables -t nat -L --line-numbers
Chain PREROUTING (policy ACCEPT)
num  target     prot opt source               destination
1    DNAT       udp  --  anywhere             anywhere             udp dpt:openvpn to:192.168.11.2:1194
2    DNAT       tcp  --  anywhere             anywhere             tcp dpt:http to:192.168.11.3:80
3    DNAT       tcp  --  anywhere             anywhere             tcp dpt:https to:192.168.11.3:443
```

**Gambar 12.** Konfigurasi *IPTables DNAT* pada PVE

Terlihat terdapat 3 (tiga) *rule* pada *IPTables* tabel NAT yang dibuat. *Rule* pertama untuk mentranslasi trafik yang menuju *server* OpenVPN yaitu dengan *port* tujuan 1194 dan IP Publik tujuan 103.217.144.161 dengan IP *Private* dari CT OpenVPN yaitu 192.168.11.2 pada *port* 1194. Sedangkan *rule* kedua dan ketiga untuk mentranslasi trafik yang mengakses layanan HTTP (port 80/tcp) dan HTTPS (port 443/tcp) ke alamat IP *Private* dari CT *reverse proxy* yaitu 192.168.11.3 pada port 80 untuk HTTP dan 443 untuk HTTPS.

### *3.7. Hasil Instalasi dan Konfigurasi pada Client Internet*

Terdapat 3 (tiga) langkah instalasi dan konfigurasi yang dilakukan pada *Client Internet* yaitu mengunduh *file profile OpenVPN* dari *server OpenVPN*, menginstalasi aplikasi *OpenVPN Client* dan meng*import file profile OpenVPN* ke dalam aplikasi *OpenVPN Client*. Apabila *file profile* telah berhasil di *import* ke aplikasi *OpenVPN Client* maka selanjutnya dapat dilakukan verifikasi koneksi ke *server OpenVPN*. Pada aplikasi *OpenVPN Client* akan tampil pesan "*Connected*" yang menandakan koneksi VPN telah berhasil dilakukan dan memperoleh alamat *IP Private* 10.8.0.x.

## 4. Pembahasan

Ujicoba terkait implementasi SDN *Simple Zone* terintegrasi *firewall* pada PVE dan *reverse proxy* pada CT "rp" meliputi penambahan masing-masing dua CT pada *zone* BRLBR dan BRLTM, koneksi ICMP antar CT pada *zone* BRLBR dan BRLTM, koneksi *Internet* dan membarui *package index* dari *apt* serta instalasi *packages* dari setiap CT pada seluruh *zone*, penghapusan seluruh CT pada *zone* BRLBR dan BRLTM.

Hasil verifikasi penambahan 4 (empat) *container* yang terhubung pada *VNet* dari 2 (dua) *zone* yaitu BRLBR dan BRLTM, seperti terlihat pada tabel 2. Terlihat terdapat 3 (tiga) komponen verifikasi yang dilakukan yaitu status pembuatan *container*, alokasi *IP Address* dari *server* DHCP dan registrasi DNS. Status pembuatan menunjukkan empat *container* telah berhasil dibuat pada *Proxmox*.

**Tabel 2.** Pembuatan Container di Zone BRLBR dan BRLTM

| Container (Zone) | Status Pembuatan Container | Alokasi IP Address dari server DHCP | Registrasi DNS |
|---|---|---|---|
| nakula (BRLBR) | Terbuat | 192.168.21.100 | Teregistrasi |
| sadewa (BRLBR) | Terbuat | 192.168.21.101 | Teregistrasi |
| yudistira (BRLTM) | Terbuat | 192.168.31.100 | Teregistrasi |
| arjuna (BRLTM) | Terbuat | 192.168.31.101 | Teregistrasi |

Selain itu setiap *container* juga telah memperoleh alokasi *IP Address* secara dinamis melalui DHCP pada bagian IPAM dari *Proxmox* SDN. Sebagai contoh terlihat salah satu *container* dengan *hostname* "nakula" memperoleh alamat IP 192.168.21.100 karena terhubung ke VNETLBR1 dengan subnet 192.168.21.0/24 pada *zone* BRLBR. Demikian pula entri RR "IN A" pada *forward lookup zone* "*idnetbrain.local*" yang terdapat di *server* DNS untuk

ke empat *container* tersebut juga telah berhasil diregistrasi. Sebagai contoh pada *zone* "*idnetbrain.local*" akan terdapat entri RR "nakula IN A 192.168.21.100" untuk *container* dengan *hostname* "nakula".

Hasil verifikasi koneksi ICMP menggunakan utilitas *ping* antar setiap CT di *Zone* BRLBR dengan BRLTM dan sebaliknya, seperti terlihat pada tabel 3.

**Tabel 3.** Verifikasi Koneksi ICMP antar CT di *Zone* BRLBR dengan BRLTM

| Container Sumber (Zone) | Container Tujuan (Zone) | Firewall Aktif Tanpa Rule Allow ICMP | Firewall Aktif Dengan Rule Allow ICMP |
|---|---|---|---|
| nakula (BRLBR) | sadewa (BRLBR) | Gagal | Reply |
| nakula (BRLBR) | yudistira (BRLTM) | Gagal | Reply |
| nakula (BRLBR) | arjuna (BRLTM) | Gagal | Reply |
| sadewa (BRLBR) | nakula (BRLBR) | Gagal | Reply |
| sadewa (BRLBR) | yudistira (BRLTM) | Gagal | Reply |
| sadewa (BRLBR) | arjuna (BRLTM) | Gagal | Reply |
| yudistira (BRLTM) | nakula (BRLBR) | Gagal | Reply |
| yudistira (BRLTM) | sadewa (BRLBR) | Gagal | Reply |
| yudistira (BRLTM) | arjuna (BRLTM) | Gagal | Reply |
| arjuna (BRLTM) | nakula (BRLBR) | Gagal | Reply |
| arjuna (BRLTM) | sadewa (BRLBR) | Gagal | Reply |
| arjuna (BRLTM) | yudistira (BRLTM) | Gagal | Reply |

Terlihat terdapat 2 (dua) komponen verifikasi yang dilakukan yaitu *firewall* aktif tanpa *rule allow* ICMP dan *firewall* aktif dengan *rule allow* ICMP di setiap *container*. Ketika *firewall* aktif namun tanpa *rule allow* ICMP maka verifikasi koneksi dengan utilitas *ping* dari setiap *container* di zone BRLBR yaitu "nakula" dan "sadewa" ke alamat IP atau FQDN dari *container* "yudistira" dan "arjuna" di *zone* BRLTM gagal. Kegagalan ini terjadi sebagai dampak *firewall* secara *default* akan menolak seluruh paket yang masuk dan tidak terdapat *rule* yang mengijinkan trafik ICMP tersebut. Sebaliknya ketika *firewall* aktif dengan *rule allow* ICMP maka verifikasi koneksi dengan utilitas *ping* dari setiap *container* di *zone* BRLBR yaitu "nakula" dan "sadewa" ke alamat IP atau FQDN dari *container* "yudistira" dan "arjuna" di zone BRLTM berhasil dilakukan yaitu ditandai dengan pesan "*Reply*". Keberhasilan tersebut terjadi sebagai dampak penambahan *rule* pada *firewall* yang mengijinkan (*allow*) trafik ICMP tersebut diterima oleh setiap *container*. Selain itu karena setiap CT menggunakan server DNS 192.168.11.2 dan juga ketersediaan entri RR "IN A" pada *forward lookup zone* "*idnetbrain.local*" yang terdapat di *server* DNS untuk ke empat *container* tersebut sehingga dapat memetakan *hostname.namadomain* ke alamat IP dari setiap CT. Sebagai contoh *sadewa.idnetbrain.local* ke alamat IP 192.168.21.101.

Hasil verifikasi pengaktifan fitur SNAT pada *VNet* di ketiga *zone* melalui 7 (tujuh) *container* yaitu *ns1*, *openvpn* dan *rp* untuk *zone* BRMTM, *nakula* dan *sadewa* untuk *zone* BRLBR serta *yudistira* dan *arjuna* untuk *zone* BRLTM, seperti terlihat pada tabel 4.

**Tabel 4.** Verifikasi Hasil Pengaktifan Fitur SNAT Pada Setiap *Container*

| Container (Zone) | Koneksi ke Google.com | Membarui package index dari apt | Instalasi packages |
|---|---|---|---|
| ns1 (BRMTM) | *Reply* | Diperbarui | Terinstalasi |
| openvpn (BRMTM) | *Reply* | Diperbarui | Terinstalasi |
| rp (BRMTM) | *Reply* | Diperbarui | Terinstalasi |
| nakula (BRLBR) | *Reply* | Diperbarui | Terinstalasi |
| sadewa (BRLBR) | *Reply* | Diperbarui | Terinstalasi |

| | | | |
|---|---|---|---|
| yudistira (BRLTM) | *Reply* | Diperbarui | Terinstalasi |
| arjuna (BRLTM) | *Reply* | Diperbarui | Terinstalasi |

Terlihat terdapat 3 (tiga) komponen verifikasi yang dilakukan dari setiap *container* tersebut yaitu koneksi ke *google.com*, membarui *package index* dari *apt* dan instalasi *package nginx* dan *links* menggunakan perintah "*apt -y install nginx links*". Khusus untuk CT "openvpn" diinstalasi *package openvpn*. Verifikasi koneksi *Internet* menggunakan utilitas *ping* dari setiap *container* ke *google.com* berhasil dilakukan yaitu ditandai dengan pesan "*Reply*". Demikian pula membarui *package index* dengan mengeksekusi perintah "*apt update*" dan menginstalasi *package nginx* sebagai *server web* serta *links* sebagai *text web browser* berhasil atau sukses dilakukan di setiap *container*. Keberhasilan verifikasi koneksi dan pengunduhan *package* dari situs di *Internet* tersebut terjadi sebagai dampak konfigurasi *recursor* pada server DNS menggunakan *PowerDNS* sehingga dapat memetakan nama *domain Internet* ke alamat IP dengan melakukan *query* ke *public DNS resolver Cloudflare* pada alamat IP 1.1.1.1 dan 1.1.1.2.

Hasil verifikasi konfigurasi *IPTables* DNAT untuk *port* 80 (HTTP) dan 443 (HTTPS) pada PVE dan *Reverse Proxy* berbasis *nginx* pada CT dengan *hostname "rp"* sehingga dapat mengakses layanan pada 4 (empat) *container* melalui *browser* dari *client Internet*, seperti terlihat pada tabel 5.

**Tabel 5.** Verifikasi Konfigurasi *IPTables* DNAT pada *Proxmox* dan *Reverse Proxy Nginx* pada CT rp

| URL Yang Diakses | Container Penyedia Layanan (*Zone*) | Hasil Pengaksesan Layanan |
|---|---|---|
| http://nakula.idnetbrain.com<br>https://nakula.idnetbrain.com | nakula (BRLBR) | Selamat Datang di nakula.idnetbrain.com |
| http://sadewa.idnetbrain.com<br>https://sadewa.idnetbrain.com | sadewa (BRLBR) | Selamat Datang di sadewa.idnetbrain.com |
| http://yudistira.idnetbrain.com<br>https://yudistira.idnetbrain.com | yudistira (BRLTM) | Selamat Datang di yudistira.idnetbrain.com |
| http://arjuna.idnetbrain.com<br>http://arjuna.idnetbrain.com | arjuna (BRLTM) | Selamat Datang di arjuna.idnetbrain.com |

Terlihat terdapat 3 (tiga) komponen yang diverifikasi yaitu URL yang diakses, *Container* Penyedia Layanan (*Zone*) dan Hasil Pengaksesan Layanan. Layanan HTTP dan HTTPS yang disediakan oleh ke empat CT yaitu nakula, sadewa pada *zone* BRLBR dan yudistira, arjuna pada *zone* BRLTM telah berhasil atau sukses diakses. Pengaksesan dilakukan menggunakan *browser Chrome* pada *client Internet* menggunakan URL yang telah di registrasi pada *server* DNS untuk domain *idnetbrain.com* yang berjalan di VPS CentOS. Sebagai contoh terlihat salah satu *container* dengan *hostname* "arjuna" memiliki konten *homepage* ketika diakses menggunakan URL http://arjuna.idnetbrain.com atau https://arjuna.idnetbrain.com adalah "Selamat Datang di arjuna.idnetbrain.com". Keberhasilan pengaksesan layanan HTTP dan HTTPS pada setiap CT di *zone* BRLBR dan BRLTM terjadi sebagai akibat penambahan *rule allow* HTTP & HTTPS pada *firewall* dari CT tersebut termasuk pada CT "rp".

Hasil verifikasi *remote access* SSH dari *Client Internet* ke setiap CT pada PVE, seperti terlihat pada Tabel 6.

**Tabel 6.** Verifikasi *Remote Access* SSH dari *Client Internet* ke CT melalui OpenVPN

| Container Tujuan (IP) | Tanpa Koneksi OpenVPN | Dengan Koneksi OpenVPN |
|---|---|---|
| ns1 (192.168.11.254) | Gagal | Sukses |
| openvpn (192.168.11.2) | Gagal | Sukses |

| Container Tujuan (IP) | Tanpa Koneksi OpenVPN | Dengan Koneksi OpenVPN |
|---|---|---|
| rp (192.168.11.3) | Gagal | Sukses |
| nakula (192.168.21.100) | Gagal | Sukses |
| sadewa (192.168.21.101) | Gagal | Sukses |
| yudistira (192.168.31.100) | Gagal | Sukses |
| arjuna (192.168.31.101) | Gagal | Sukses |

Hasil verifikasi penghapusan 4 (empat) *container* yang terhubung pada *VNet* dari 2 (dua) *zone* yaitu BRLBR dan BRLTM, seperti terlihat pada tabel 7.

**Tabel 7.** Penghapusan *Container* di *Zone* BRLBR dan BRLTM

| Container (*Zone*) | Status Penghapusan Container | IPAM | DNS |
|---|---|---|---|
| nakula (BRLBR) | Terhapus | Terhapus | Terhapus |
| sadewa (BRLBR) | Terhapus | Terhapus | Terhapus |
| yudistira (BRLTM) | Terhapus | Terhapus | Terhapus |
| arjuna (BRLTM) | Terhapus | Terhapus | Terhapus |

Terlihat terdapat 3 (tiga) komponen verifikasi yang dilakukan yaitu status penghapusan *container*, IPAM dan DNS. Status penghapusan menunjukkan ke empat *container* telah berhasil dihapus pada PVE. Demikian pula entri alokasi alamat IP pada bagian IPAM dari SDN PVE dan entri RR "IN A" pada *forward lookup zone* "*idnetbrain.local*" yang terdapat di *server* DNS untuk ke empat *container* tersebut juga telah berhasil dihapus. Keseluruhan ujicoba terkait penerapan SDN pada PVE yang dibangun menggunakan VPS tersebut dilakukan melalui komputer *client* yaitu sebuah *laptop* dengan sistem operasi *Windows* 11 yang dikoneksikan ke *Internet*. Ujicoba dilakukan sebanyak 10 (sepuluh) kali untuk setiap komponen. *Browser Chrome* digunakan untuk mengujicoba fitur *reverse proxy* dengan mengakses URL dari layanan HTTP/HTTPS yang telah aktif di setiap CT. Sedangkan aplikasi *OpenVPN Connect* digunakan untuk mengujicoba konektivitas ke *server* VPN. Setelah terhubung ke VPN maka dilakukan ujicoba *remote access* menggunakan aplikasi *Putty SSH Client* ke setiap CT. Rangkuman dari keseluruhan hasil ujicoba per komponen tersebut disajikan dalam bentuk tabel.

## 5. Kesimpulan

Adapun kesimpulan yang dapat diambil adalah SDN bertipe *simple zone* pada PVE dapat digunakan untuk mengontrol konfigurasi *virtual network* secara terpusat dan lebih sederhana seperti *routing*, DHCP, SNAT, registrasi *hostname* dan IP dari CT ke *forward lookup zone* di server DNS. Pengaktifan *firewall* dan pembuatan *rule* di level *cluster* dan CT dari PVE serta OpenVPN dapat memproteksi infrastruktur ketika diakses baik dari internal maupun eksternal. Sedangkan penerapan *nginx reverse proxy* dapat mengamankan akses layanan HTTP/HTTPS pada CT di PVE. Namun SDN pada PVE versi 8.2.4 masih memiliki beberapa kekurangan antara lain belum memiliki *Graphical User Interface (GUI)* untuk menyesuaikan DHCP *options* terkait DNS sehingga dilakukan pengaturan secara manual pada *file* konfigurasi *simple zone*. Fitur *DNS Zone Reverse* dari SDN tidak menambahkan entri *reverse lookup* di *server* DNS meskipun telah diatur pada *simple zone*. Selain itu SDN belum mendukung fitur DNAT sehingga dilakukan konfigurasi *IPTables* secara manual dan disimpan permanen melalui pemanfaatan *iptables-persistent*. Saran untuk pengembangan penelitian ini lebih lanjut adalah mengintegrasikan SDN dengan *Intrusion Prevention System (IPS)* agar sistem dapat mendeteksi dan memitigasi ancaman dengan menginspeksi aliran trafik jaringan secara *real time* sehingga memastikan komunikasi antar elemen jaringan tetap aman atau terproteksi.

## Referensi


[1] A. Kurniawan Yusuf, A. Hendri Hendrawan, and Y. Afrianto, "Building Virtual Private Server In Net-Centric Computing Laboratory," *Jurnal Teknik Informatika C.I.T*, vol. 11, no. 2, 2019, [Online]. Available: www.medikom.iocspublisher.org/index.php/JTI

[2] İ. Yoşumaz, "An Examination of Cyber Security Solutions in Public and Private IaaS Infrastructures," *International Journal of Information Security Science*, vol. 13, no. 3, pp. 1–29, Sep. 2024, doi: 10.55859/ijiss.1475423.

[3] M. Kondoj, H. Langi, Y. Putung, and V. Lengkong, "Performance Analysis of Cloud Computing Based E-Commerce Server Using PROXMOX Virtual Environment," INSTICC, Dec. 2023, pp. 741–745. doi: 10.5220/0011876000003575.

[4] A. Kholid, A. Faif, P. Hatta, and E. S. Wihidayat, "Performance Analysis of Proxmox and Virtualbox with Overhead and Linearity Parameters to Support Server Administration Practice," *Journal of Informatics and Vocational Education (JOIVE)*, vol. 7, no. 2, pp. 35–41, 2024.

[5] Y. Ariyanto, "Single Server-Side And Multiple Virtual Server-Side Architectures: Performance Analysis On Proxmox VE For Elearning Systems," *Journal of Engineering and Technology for Industrial Applications*, vol. 9, no. 44, pp. 25–34, Dec. 2023, doi: 10.5935/jetia.v9i44.903.

[6] R. Achmad Alfarizhi, T. Ariyadi, and M. Ulfa, "Implementasi Prototype Bisnis IT Layanan VPS Dan Web Hosting Sebagai Laboratorium Research Universitas Bina Darma," *Jurnal INOVTEK POLBENG - Seri Informatika*, vol. 9, no. 2, pp. 526–538, 2024.

[7] V. Oleksiuk and O. Oleksiuk, "The practice of developing the academic cloud using the Proxmox VE platform," *Educational Technology Quarterly*, vol. 2021, no. 4, pp. 605–616, Dec. 2021, doi: 10.55056/etq.36.

[8] Y. Ariyanto, B. Harijanto, V. A. H. Firdaus, and S. N. Arief, "Performance analysis of Proxmox VE firewall for network security in cloud computing server implementation," in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, Jan. 2020. doi: 10.1088/1757-899X/732/1/012081.

[9] Baharuddin, D. Ampera, H. Fibriasari, M. A. R. Sembiring, and A. Hamid, "Implementation of cloud computing system in learning system development in engineering education study program," *International Journal of Education in Mathematics, Science and Technology*, vol. 9, no. 4, pp. 697–740, 2021, doi: 10.46328/ijemst.2114.

[10] S. Mahipal and V. Ceronmani Sharmila, "Virtual Machine Security Problems and Countermeasures for Improving Quality of Service in Cloud Computing," in *Proceedings - International Conference on Artificial Intelligence and Smart Systems, ICAIS 2021*, Institute of Electrical and Electronics Engineers Inc., Mar. 2021, pp. 1319–1324. doi: 10.1109/ICAIS50930.2021.9395922.

[11] A. M. Abdelrahman *et al.*, "Software-defined networking security for private data center networks and clouds: Vulnerabilities, attacks, countermeasures, and solutions," *International Journal of Communication Systems*, vol. 34, no. 4, Mar. 2021, doi: 10.1002/dac.4706.

[12] M. Rahouti, K. Xiong, Y. Xin, S. K. Jagatheesaperumal, M. Ayyash, and M. Shaheed, "SDN Security Review: Threat Taxonomy, Implications, and Open Challenges," 2022, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2022.3168972.

[13] Y. Zheng, Z. Li, X. Xu, and Q. Zhao, "Dynamic defenses in cyber security: Techniques, methods and challenges," *Digital Communications and Networks*, vol. 8, no. 4, pp. 422–435, Aug. 2022, doi: 10.1016/j.dcan.2021.07.006.

[14] S. * Balachandran, Dominic, and J. Sivankalai, "A Comparative Analysis of VPN and Proxy Protocols in Library Network Management," *Library Progress International*, vol. 44, no. 3, pp. 17006–17020, 2024, Accessed: Nov. 20, 2024. [Online]. Available: https://bpasjournals.com/library-science/index.php/journal/article/view/2747

[15] C. H. Chua and S. C. Ng, "Open-Source VPN Software: Performance Comparison for Remote Access," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Aug. 2022, pp. 29–34. doi: 10.1145/3561877.3561882.

[16] 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," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Jul. 2021. doi: 10.1088/1742-6596/1908/1/012026.