SIMULASI JARINGAN UNTUK SISTEM TERDISTRIBUSI SEAFILE DENGAN GNS3

Jessen Chayadi [1] Finnia Li [2] Richard Christian [3] Nelson [4] Valentino Almendo Radjawane [5]

[1) 2) 3) 4) 5)]Teknik Informatika Universitas Tarumanagara
Jl. Let. Jend. S. Parman No. 1, Jakarta 11440 Indonesia
email : [1)]jessen.535220023@stu.untar.ac.id, [2)] finnia.535220030@stu.untar.ac.id, [3)] richard.535220018@stu.untar.ac.id, [4)] nelson.535220021@stu.untar.ac.id, [5)] valentino.535220040@stu.untar.ac.id

ABSTRACT

Setiap perusahaan pasti membutuhkan sistem terdistribusi yang dapat menyimpan data mereka dan menghubungkan data dari setiap karyawan atau setiap cabang. Maka dari itu, disini kami membuat sebuah simulasi sistem terdistribusi yang dapat diakses oleh karyawan dari setiap cabang di sebuah perusahaan dengan menggunakan sistem terdistribusi Seafile dengan menggunakan GNS3. Sistem terdistribusi ini menggunakan layanan Seafile untuk membuat seorang karyawan dapat mengakses data perusahaan dari sebuah website. Sistem ini akan meningkatkan kolaborasi dan efisiensi manajemen file bagi perusahaan. Dengan memanfaatkan layanan Seafile, kami memberdayakan setiap karyawan untuk mengakses dan berbagi file melalui antarmuka web yang mudah digunakan. Selain itu, kami membuat sistem ini dapat mendekati dan meningkatkan aksesibilitas, fleksibilitas, dan lebih efektif untuk kolaborasi. Simulasi ini bertujuan untuk menganalisa kinerja dan estimasi keandalan dan skalabilitas sistem. Simulasi ini juga fokus dalam menunjukkan efisiensi jaringan dan aksesibilitas jaringan yang akan berkontribusi untuk meningkatkan kolaborasi, kerjasama, dan manajemen file. Selain itu, proyek kami juga berfokus pada keamanan dengan menerapkan koneksi aman yang aman untuk transfer data dan berbagi data.

Kata Kunci

GNS3, Seafile, Ubuntu, Linux, Distributed system.

1. Pendahuluan

Semakin besar sebuah perusahaan maka semakin sulit untuk mengendalikan data-data yang ada didalamnya. Kolaborasi antar pegawai untuk manajamen data menjadi tantangan baru. Karya ilmiah ini fokus merancang sistem distribusi berbasis web menggunakan Seafile. Tujuan dibuatnya system ini bertujuan untuk meningkatkan efisiensi penyimpanan data, memperkuat keamanan akses, dan meningkatkan sistem yang mudah diakses oleh berbagai perangkat.

Sebelumnya, beberapa platform system terdistribusi telah teruji, tapi terus menerus ditemui masalah dengan keamanan dan akses terdistribusi[1]. Karya ilmiah ini diharapkan dapat memberikan solusi dalam untuk masalah-masalah tersebut dan memberikan landasan yang kuat untuk pengembangan sistem terdistribusi berbasis web yang lebih baik dan efisien di masa mendatang.

Seafile adalah platform penyimpanan data terdistribusi yang memungkinkan kolaborasi dan berbagi file dengan efisien. Dalam beberapa penelitian sebelumnya, seperti yang dijelaskan oleh Zhang., Seafile menunjukkan performa yang unggul dalam hal kecepatan akses dan keamanan data dibandingkan dengan beberapa platform penyimpanan lainnya [6]. Selain itu, Penelitian juga menggarisbawahi kelebihan Seafile dalam manajemen akses terdistribusi dan integrasi dengan berbagai perangkat [7].

Untuk mendukung implementasi dan pengujian sistem ini, simulasi jaringan menggunakan GNS3 (Graphical Network Simulator-3) akan digunakan. GNS3 adalah alat simulasi jaringan yang memungkinkan pengguna untuk merancang, menguji, dan memecahkan masalah topologi jaringan secara virtual[9]. GNS3 telah terbukti menjadi alat yang efektif dalam simulasi jaringan dan analisis performa sistem terdistribusi [8][10].

Dengan demikian, penelitian ini tidak hanya berfokus pada peningkatan efisiensi dan keamanan dalam manajemen data, tetapi juga pada penggunaan alat simulasi yang tepat untuk menguji dan mengoptimalkan sistem tersebut.

2. Studi Pustaka

2.1 Jaringan dan Keamanan Komputer

Jaringan komputer merupakan beberapa perangkat komputasi yang terhubung secara elektronik untuk berbagai data yang dapat juga memungkinkan untuk saling berkomunikasi. Tujuan dari jaringan komputer ini sendiri agar setiap jaringan dapat bertukar data dan berbagi sumber daya satu sama lain.

Dalam jaringan antar komputer diperlukan pengolahan jaringan yang baik agar jaringan tersebut tetap terhubung dan salah satu permasalahan dari

pengelolaan jaringan merupakan keamanan antar jaringan.

Sistem keamanan jaringan komputer merupakan hal yang penting didalam pembangunan jaringan. Pembangunan jaringan masih banyak menggunakan router yang memiliki *system firewall* yang terintegrasi. Keamanan jaringan juga dapat dikontrol dengan menyesuaikan *network sharing properties* yang dapat membatasi *folder* dan *file* yang dapat dilihat oleh pengguna tertentu didalam sistem jaringan. Didalam jaringan dan keamanan komputer juga berkaitan erat dengan pembagian jaringan yang dikenal dengan *subnetting*, protocol yang menghubungkan antar jaringan komputer atau yang dikenal NAT, *Transport Protocol* (TCP/UDP) dan *Application Services* (*Application Layer*), VPN, dan system yang membatasi akses ke jaringan dan system atau yang dikenal dengan *Firewall*. [1]

2.1.1 *Subnetting*

Subnetting menciptakan atau membagi beberapa jaringan tambahan tanpa mengurangi maksimum *host* yang ada dalam tiap jaringan. *Subnetting* merupakan jenis IP *Address* yang digunakan perangkat dengan jaringan berskala lokal atau dikatakan LAN dan IP *Address* ini tidak dikenal pada jaringan internet global. [1]

Subnetting sendiri memiliki beberapa kegunaan yang diantaranya:

1. Mengefisienkan alokasi alamat IP
2. Memperbesar skala jaringan
3. Memaksimalkan penggunaan alamat IP

Untuk *subnet* dalam *subnetting* tidak dapat berkomunikasi secara langsung tanpa bantuan *router* dan ditujukan untuk menghemat pemakaian IP *Address* dan *subnetting* memiliki dua metode yaitu:

a) *Static Length Subnet Mask Method* (SLSM)

Static Length Subnet Mask (SLSM) adalah metode subnetting yang membagi sebuah jaringan menjadi lebih banyak subnet yang masing-masing memiliki panjang yang sama. Dan untuk menghitung subnetting dengan metode SLSM dapat menggunakan rumus $2n \geq$ subnet yang dimana n akan menjadi jumlah bit yang dibutuhkan untuk tambahan 24 bits dari original network address.

b) *Variable Length Subnet Mask Method* (VLSM)

Variable Length Subnet Mask (VLSM) adalah metode subnetting yang memungkinkan administrator jaringan untuk membagi ruang alamat IP ke subnet dengan ukuran yang berbeda. VLSM menyesuaikan panjang subnet mask dengan jumlah host yang ada di setiap subnet. Dan untuk menghitung subnetting dengan metode VLSM dapat menggunakan rumus $2n - 2 \geq$ subnet yang dimana n akan menjadi jumlah bit untuk identitas host dan berlanjut untuk subnet selanjutnya. [2]

Tabel 1. Contoh *subnetting* dengan metode VLSM

Nama	Host	Net Addr	Range IP	Subnet Mask	Broadcast
Host1	50	192.168.17.0	192.168.17.1 - 192.168.17.62	255.255.255.192	192.168.17.63
Host2	10	192.168.17.64	192.168.17.65 - 192.168.17.78	255.255.255.240	192.168.17.79
Host3	10	192.168.17.80	192.168.17.81 - 192.168.17.94	255.255.255.240	192.168.17.95
Host4	8	192.168.17.96	192.168.17.97 - 192.168.17.110	255.255.255.240	192.168.17.111
Host5	8	192.168.17.112	192.168.17.113 - 192.168.17.126	255.255.255.240	192.168.17.127
Host6	8	192.168.17.128	192.168.17.129 - 192.168.17.142	255.255.255.240	192.168.17.143

2.1.2 *Network Address Translation* (NAT)

NAT (*Network Address Translation*) merupakan proses menghubungkan lebih dari satu komputer ke jaringan internet dengan menggunakan satu alamat IP dengan mengubah alamat sumber atau tujuan di *header* IP paket saat sedang dalam perjalanan.

Dengan jaringan yang menggunakan alamat lokal (*private*) yang dibuat NAT, kita dapat berkomunikasi ke internet dengan satu IP yang mewakili sekelompok user yang dialokasikan oleh ISP.

NAT dibuat untuk menyelesaikan masalah alamat Internet IPv4 yang terbatas. NAT terjadi ketika beberapa perangkat memerlukan akses Internet tetapi ISP telah menetapkan hanya satu alamat Internet IPv4 yang sudah ditetapkan oleh Penyedia Layanan Internet (ISP). [3]

2.1.3 *Transport Protocol* dan *Application Service*

TCP (*Transmission Control Procotol*) adalah salah satu teknologi yang sangat bermanfaat dalam dunia jaringan komputer. Setiap komputer saat ini menggunakan protokol ini untuk mengirimkan dan menerima data, dan memiliki kelebihan dalam membaca data dengan cepat, yang memungkinkan pengiriman data dengan lebih efisien. Salah satu karakteristik TCP adalah orientasinya dalam mengutamakan koneksi. Sebelum protokol TCP dapat mengirimkan data, protokol ini melakukan sesi koneksi. Jika terjadi gangguan atau ketidaklancaran dalam koneksi, TCP tidak dapat diandalkan untuk mengirimkan pesan sesuai kebutuhan. Sedangkan UDP adalah *Protokol Transport Layer* yang tidak dapat diandalkan (UDP) yang merupakan kebalikannya dari *transport layer* TCP. UDP tidak mengirimkan keterangan atau pengakuan meskipun pengirim data gagal, sehingga sangat mungkin data rusak saat dikirim. Penggunaan protokol UDP memang diperlukan jika Anda melakukan konektivitas yang membutuhkan kecepatan tanpa perlu mengkhawatirkan keutuhan data pada saat transfer data antar *host*. [4]

Didalam TCP data akan dibagi menjadi bagian yang lebih kecil sesuai dengan *bandwidth* atau frekuensi pengiriman. Pada lapisan TCP, data akan dikemas dengan

informasi *header* yang diperlukan, sehingga data akan disusun kembali ketika sampai pada tujuan. Setelah data dikemas dengan *header* TCP, lapisan IP menerima data dari TCP dan menambahkan *header*nya sendiri ke dalam data. Setelah itu, IP meneruskan data ke tujuannya.

Application layer adalah lapisan tertinggi dalam model *Open Systems Interconnection* (OSI) *reference*. Lapisan ini menyediakan antarmuka antara protokol jaringan dan aplikasi, dan juga berfungsi sebagai tempat di mana pengguna berinteraksi dengan aplikasi yang menggunakan fitur jaringan. *Application layer* sendiri sebagai penyedia layanan bagi sebuah aplikasi, penyedia layanan seperti *interface* untuk *File Transfer Protocol* (FTP) yang dimana aplikasi memiliki keterkaitan dengan jaringan. [4]

2.1.4 *Virtual Private Network* (VPN)

VPN (*Virtual Private Network*) adalah jaringan pribadi yang dikonfigurasi dalam jaringan publik, seperti jaringan operator atau internet, untuk memanfaatkan kapasitas manajemen jaringan yang besar dan ukuran ekonomi. Bisnis sering menggunakan VPN untuk membangun jaringan area luas yang mencakup area geografis yang luas, memberikan koneksi situs-ke-situs ke kantor cabang, dan memungkinkan pengguna seluler menghubungi LAN perusahaan mereka. Dari perspektif penyedia, banyak pelanggan menggunakan fasilitas jaringan publik bersama-sama, dan lalu lintas setiap pelanggan terpisah dari lalu lintas lainnya. Lalu lintas VPN hanya dapat bepergian dari sumber VPN ke tujuan dalam VPN yang sama. Seringkali, fasilitas enkripsi dan otentikasi VPN tersedia. [2]

2.1.5 *Firewall*

Firewall adalah metode yang sangat berguna dan penting dalam mengamankan jaringan, dan merupakan suatu model atau sistem mekanisme yang diterapkan pada perangkat *hardware*, *software*, dan sistem itu sendiri dengan tujuan untuk melindungi, baik dengan menyaring, membatasi, atau bahkan menolak suatu atau semua hubungan kegiatan segmen tertentu pada jaringan pribadi dengan jaringan luar yang tidak terkait dengan segmen tersebut.

Firewall digunakan untuk membatasi atau mengontrol akses pihak luar ke jaringan pribadi. Saat ini, *firewall* menjadi istilah umum yang mengacu pada sistem yang mengatur hubungan antara dua jenis jaringan. Saat ini, banyak bisnis memiliki akses ke internet dan jaringan berbadan hukum, jadi penting untuk melindungi perangkat digital mereka dari para peretas, pemata-mata, dan pencuri data lainnya. [5]

2.2 Aplikasi Terdistribusi

Seafile dan Ubuntu *Virtual Machine* merupakan aplikasi yang digunakan dalam perancangan penyimpanan terdistribusi ini. Seafile adalah platform kolaborasi berbasis file yang memungkinkan pengguna untuk menyimpan, berbagi, dan mengelola data secara terdistribusi. Dengan menggunakan Seafile, pengguna dapat membuat repositori dan berkolaborasi secara efisien melalui sinkronisasi file yang cepat dan aman. Seafile memiliki banyak fitur yang membuatnya unggul sebagai platform penyimpanan terdistribusi, [6]

Pertama, model penyimpanan terdistribusi memungkinkan file disimpan di berbagai *node server*, memastikan keberlanjutan dan ketersediaan tinggi. Fitur sinkronisasi file yang sangat cepat memungkinkan perubahan dilakukan pada satu perangkat segera diteruskan ke perangkat lainnya, yang menghasilkan pengalaman kerja yang sinergis dan efisien. Dengan manajemen versi *file* otomatis Seafile, pengguna dapat dengan mudah melacak dan mengelola evolusi *file*. Fokus utama adalah keamanan, dengan dukungan enkripsi *end-to-end* untuk mencegah akses yang tidak sah ke data. Melalui fiturnya, seperti komentar, diskusi, dan pembaruan *real-time*, Seafile juga membantu tim bekerja sama yang bisa meningkatkan produktivitas tim. [6]

Gambar 1. Interface Seafile

Ubuntu *Virtual Machine* digunakan sebagai sistem operasi utama yang memanfaatkan stabilitas, keamanan, dan dukungan komunitas yang kuat dari sistem operasi. Penggunaan *virtual machine* juga dimaksudkan untuk menjaga lingkungan proyek terpisah dari sistem operasi *host*, memungkinkan pengelolaan sumber daya yang efektif, dan menjaga proyek terpisah dari sistem operasi host. Web server proyek ini dikembangkan secara internal dengan memanfaatkan layanan Apache sebagai fondasi utama. Apache dipilih karena reputasinya sebagai *server* web yang terpercaya dan memiliki dukungan fitur yang kaya untuk aplikasi web. Keuntungan penggunaan Apache termasuk kinerja tinggi, kemudahan konfigurasi, dan dukungan untuk modul ekstensif yang meningkatkan fungsionalitas *server* [7].

3. Hasil Percobaan

3.1 Instalasi dan Pengaturan

Simulasi topologi jaringan dibuat dalam GNS3, dimana dalam topologi jaringan ini terdiri dari tiga lapisan yaitu *INSIDE*, *OUTSIDE*, dan DMZ (*Demilitarized*

Zone). *INSIDE*, *OUTSIDE*, dan DMZ sendiri dibedakan dari *security level* yang di set pada Cisco ASAv.

INSIDE memiliki *security level* tertinggi yaitu 100. Dimana *INSIDE* biasanya merujuk kepada jaringan inti dalam perusahaan dan menyimpan data-data sensitive yang biasanya digunakan oleh *internal* perusahaan serta sangat terbatas penggunanya. *OUTSIDE* memiliki *security level* terendah bahkan tidak memiliki *security level* sama sekali karena merupakan bagian yang terhubung dengan internet dan dapat diakses langsung oleh area di luar jaringan. Sedangkan pada DMZ memiliki *security level* 50 dan berada diantara *INSIDE* serta *OUTSIDE*. DMZ biasanya berisi layanan *web server* yang tidak langsung terhubung ke *internal.*

Rangkaian simulasi dirangkai sedemikian rupa dalam bentuk topologi jaringan pada GNS3.

Gambar 2. Simulasi topologi jaringan menggunakan GNS3

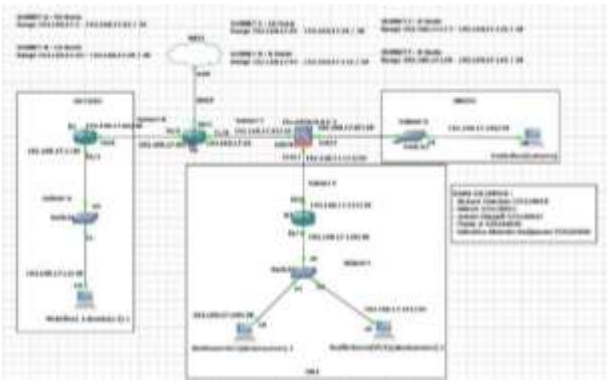
Berikut merupakan hasil *subnetting* dan alokasi IP *Address* pada masing masing *device* yang ada pada *layout*.

Tabel 2. Rangkuman hasil *subnetting* dan alokasi IP ke *device*

Subnet	Net Addr	Host	Range	Broadcast	Device
A	192.168.17.0	50	192.168.17.1/26 – 192.168.17.62/26	192.168.17.63	WebClient (eth0/0) (192.168.17.12/26) dan R1 (eth0/1) (192.168.17.1/26)
B	192.168.17.64	10	192.168.17.65/28 – 192.168.17.78/28	192.168.17.79	R1 (eth0/0) (192.168.17.66/28) dan R2 (eth0/0) (192.168.17.65/28)
C	192.168.17.80	10	192.168.17.81/28 – 192.168.17.94/28	192.168.17.95	R2 (eth1/0) (192.168.17.81/28) dan Cisco ASAv (eth0/0) (192.168.17.82/28)
D	192.168.17.96	8	192.168.17.97/28 – 192.168.17.110/28	192.168.17.111	Cisco ASAv (eth0/2) (192.168.17.97/28) dan InsideHost (eth0/0) (192.168.17.100/28)
E	192.168.17.112	8	192.168.17.113/28 – 192.168.17.126/28	192.168.17.127	Cisco ASAv (eth0/1) (192.168.17.113/28) dan R3 (eth0/0) (192.168.17.114/28)
F	192.168.17.128	8	192.168.17.129/28 – 192.168.17.142/28	192.168.17.143	R3 (eth0/1) (192.168.17.129/28), WebServer (eth0/0) (192.168.17.140/28), dan SeafileServer (eth0/0) (192.168.17.141/28)

Langkah-langkah *setting* yang digunakan pada simulasi topologi jaringan yaitu:

1. Menambahkan *ip address* sesuai *interface* masing-masing pada *router*, *virtual machine*, dan khusus pada Cisco ASAv ada penambahan *security level* pada setiap lapisan yaitu *INSIDE*, *OUTSIDE*, dan DMZ.
2. Menambahkan *ip route* untuk *static routing* pada setiap *router* dan *Cisco ASAv* agar dapat terhubung ke internet.
3. Menyetting DNS server yaitu 8.8.8.8 dan 8.8.4.4 pada Cisco *Router*, Cisco ASAv, dan *Virtual Machine.*
4. Set NAT pada *router* R2 untuk akses internet dengan konfigurasi *access list* pada setiap *subnet* nya.
5. Set NAT dan *Access Control List* pada Cisco ASAv dengan cara memasukkan *subnet* ke dalam *object network* sesuai lapisannya, contoh : *object network inside-subnet*.
6. *Test* ping dari satu *router* ke *router* lainnya, setiap *router* ke VM, VM ke *gateway*, VM ke 8.8.8.8, dan terakhir dari VM ke google.com.
7. Instalasi Seafile pada Ubuntu VM sebagai Seafile server dengan mengikuti tutorial dari link https://manual.seafile.com/deploy/using_mysql/ .
8. Setelah langkah-langkah diatas dilakukan maka *WebClient*, *WebServer*, dan *Storage Server*



menggunakan Seafile sudah bisa digunakan.

3.2 Hasil Simulasi

Berdasarkan topologi jaringan diatas maka didapat *routing table* untuk setiap *router* sebagai berikut :

a) *Router* R1

Tabel 3. Routing Table pada Router R1

Network Address	Gateway	Interface
192.168.17.0/26	Connected	eth0/1
192.168.17.64/28	Connected	eth0/0
192.168.17.80/28	192.168.17.65	eth0/0
192.168.17.96/28	192.168.17.65	eth0/0
192.168.17.112/28	192.168.17.65	eth0/0
192.168.17.128/28	192.168.17.65	eth0/0

b) *Router* R2

Tabel 4. Routing Table pada Router R2

Network Address	Gateway	Interface
192.168.17.0/26	192.168.17.66	eth0/0
192.168.17.64/28	Connected	eth0/0
192.168.17.80/28	Connected	eth1/0
192.168.17.96/28	192.168.17.82	eth1/0
192.168.17.112/28	192.168.17.82	eth1/0
192.168.17.128/28	192.168.17.82	eth1/0

c) *Router* R3

Tabel 5. Routing Table pada Router R3

Network Address	Gateway	Interface
192.168.17.0/26	192.168.17.113	eth0/0
192.168.17.64/28	192.168.17.113	eth0/0
192.168.17.80/28	192.168.17.113	eth0/0
192.168.17.96/28	192.168.17.113	eth0/0
192.168.17.112/28	Connected	eth0/0
192.168.17.128/28	Connected	eth0/1

d) Cisco ASAv

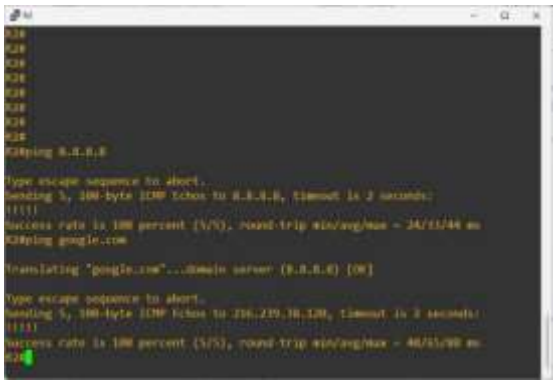
Tabel 6. Routing Table pada Cisco ASAv

Network Address	Gateway	Interface
192.168.17.0/26	192.168.17.81	eth0/0
192.168.17.64/28	192.168.17.81	eth0/0
192.168.17.80/28	Connected	eth0/0
192.168.17.96/28	Connected	eth0/2
192.168.17.112/28	Connected	eth0/1
192.168.17.128/28	192.168.17.114	eth0/1

Selanjutnya untuk skenario pengujian koneksi jaringan akan dijelaskan sesuai dengan langkah-langkah sebagai berikut :

1. Menguji konektivitas semua *device* dengan internet dengan cara ping 8.8.8.8 dan google.com.

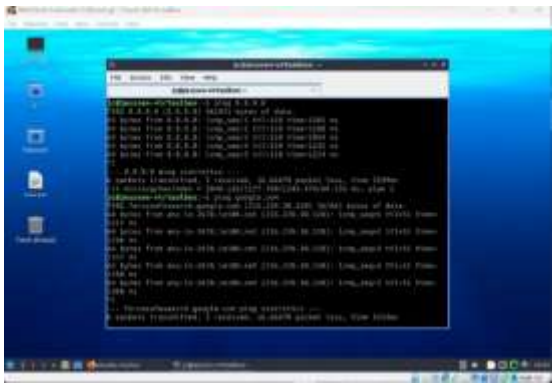
Gambar 3. Menguji konektivitas internet pada *Router* R1



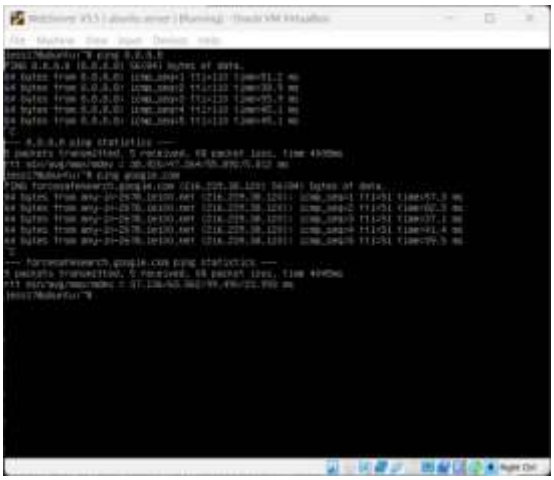
Gambar 4. Menguji konektivitas internet pada *Router* R2

Gambar 5. Menguji konektivitas internet pada *Router* R3

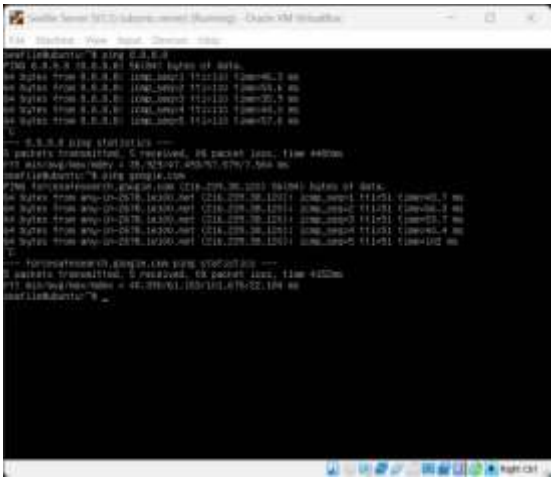
Gambar 6. Menguji konektivitas internet pada Cisco ASAv



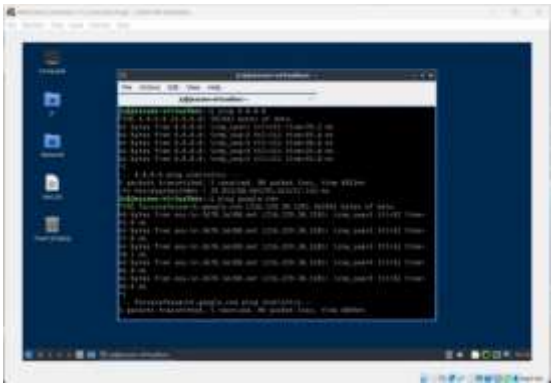
Gambar 7. Menguji konektivitas internet pada *host* yang berada di *outside zone*



Gambar 8. Menguji konektivitas internet pada *WebServer* yang berada di *demilitarized zone*



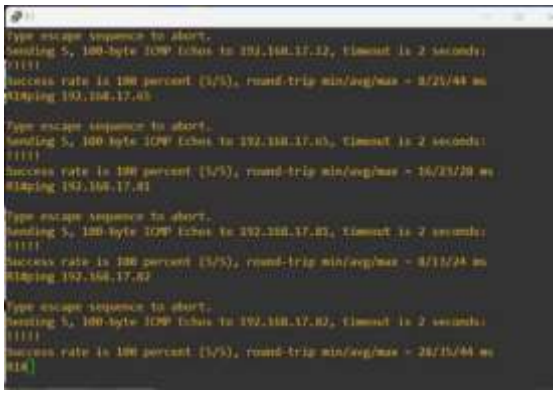
Gambar 9. Menguji konektivitas internet pada Seafile Server yang berada di *demilitarized zone*



Gambar 10. Menguji konektivitas internet pada *host* yang berada di *inside zone*

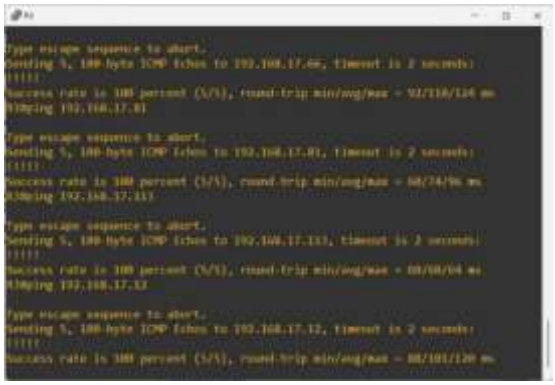
Seperti yang dapat dilihat pada Gambar 3 sampai dengan Gambar 10, semua *host* sudah terkoneksi kepada NAT melalui *static routing* sehingga sudah dapat mengakses internet.

2. Menguji konektivitas antar semua router dan Cisco ASAv

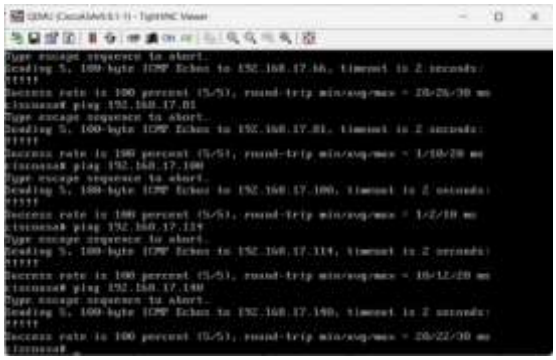


Gambar 11. Menguji konektivitas antar *host* dari *Router* R1

Gambar 12. Menguji konektivitas antar *host* dari *Router* R2



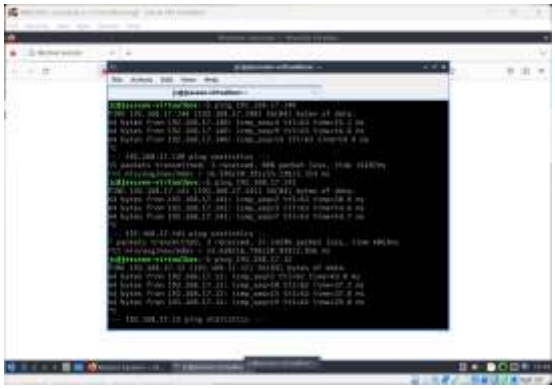
Gambar 13. Menguji konektivitas antar *host* dari *Router* R3



Gambar 14. Menguji konektivitas antar host dari Cisco ASAv

Berdasarkan Gambar 11 sampai Gambar 14, telah dilakukan uji konektivitas antar sesama *router* dan beberapa *host* lainnya di setiap *router* yaitu R1, R2, R3, dan Cisco ASAv.

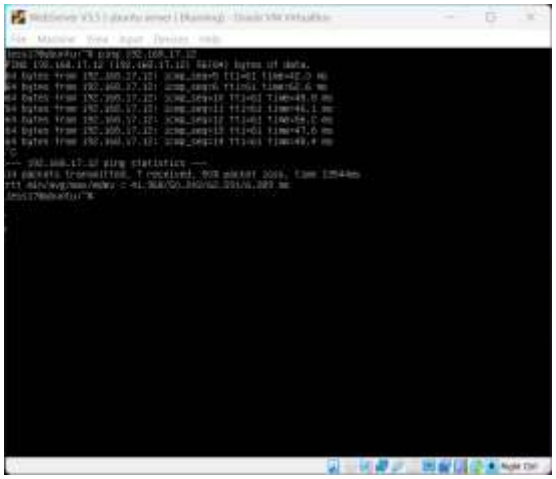
3. Menguji konektivitas *client* yang berada di *inside zone* ke *demilitarized zone* dan *outside zone*.



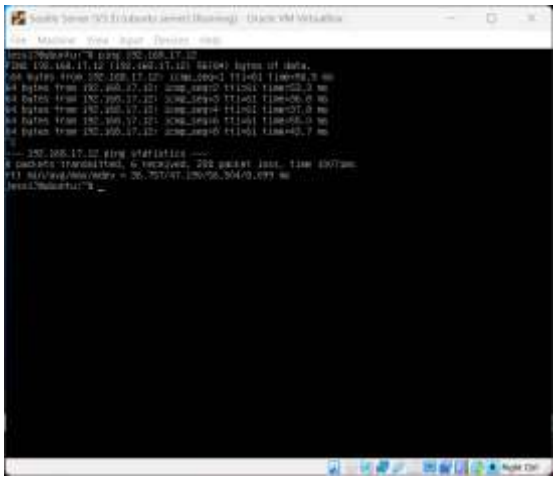
Gambar 15. Menguji konektivitas dari client yang terdapat di *inside zone* ke *server* yang berada di *demilitarized zone* dan *client* yang berada di *outside zone*

Disini dilakukan pengujian konektivitas antara *client* yang berada pada *inside zone* dengan kedua *server* yang berada di *demilitarized zone* dan *client* yang berada di *outside zone* dengan cara ping *host* nya, meskipun terdapat beberapa *packet loss* yang terjadi tapi koneksi tetap berhasil terhubung seperti pada Gambar 15.

4. Menguji konektivitas *host* yang berada di *demilitarized zone* ke *client* yang berada di *outside zone*.



Gambar 16. Menguji konektivitas dari *WebServer* ke *WebClient* yang berada di *outside zone*

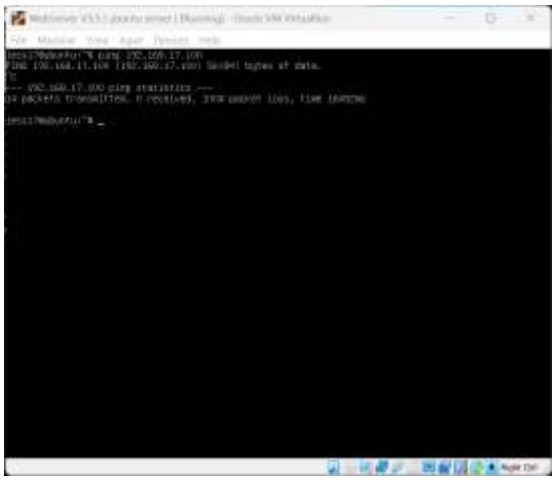


Gambar 17. Menguji konektivitas dari Seafile Server ke *client* yang berada di *outside zone*

Disini dilakukan pengujian apakah dari *server* yang berada pada *demilitarized zone* sudah terkoneksi dengan *client* yang berada pada *outside zone*, dan seperti pada Gambar 16 dan Gambar 17, *server* berhasil terhubung dengan *client* yang ditandai dengan suksesnya ping terhadap *client* tersebut.

5. Membuktikan bahwa *host* yang berada pada zona yang memiliki *security level* lebih rendah tidak dapat mengakses zona yang memiliki *security level* yang lebih tinggi.

Gambar 18. *Client* yang berada di *outside zone* tidak bisa ping ke *server* yang berada di *demilitarized zone*, maupun *client* yang berada pada *inside zone*

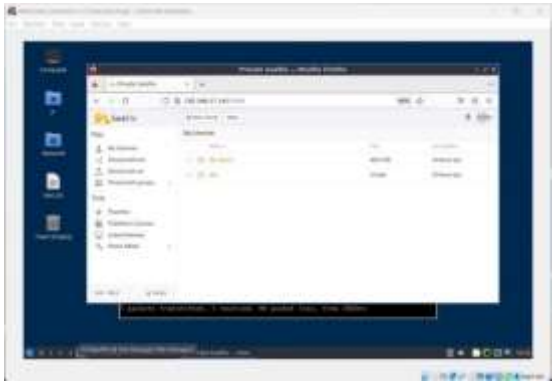


Gambar 19. *Server* yang berada di *demilitarized zone* pun tidak bisa mengakses *client* yang berada di *inside zone*

Pada Gambar 18 dan Gambar 19, telah dibuktikan bahwa *zone* yang memiliki *security level* lebih rendah tidak akan bisa mengakses *zone* yang memiliki *security level* yang lebih tinggi secara langsung tanpa konfigurasi lebih lanjut. Misalnya, *outside zone* (*security-level* 0) tidak dapat mengakses DMZ (*security-level* 50) maupun *inside zone* (*security-level* 100).

6. Mengakses *WebServer* dan Seafile Server melalui *client* yang berada pada *inside zone*.

Gambar 20. Mengakses *Web Server* (192.168.17.140) dari *client* yang berada pada *inside zone*.



Gambar 21. Mengakses Seafile Server (192.168.17.141:8000) dari *client* yang berada pada *inside zone*.

Disini kedua *server* diakses secara langsung melalui *client* yang berada pada *inside zone* seperti pada Gambar 20 dan Gambar 21, dikarenakan *inside zone* memliki *security level* lebih tinggi yaitu 100, maka *client* dapat langsung mengakses *server* yang berada pada *demilitarized zone* secara langsung.

7. *WebClient* yang berasal dari *outside zone* mengakses *WebServer* yang berada pada *demilitarized zone* melalui *public* ip yang sudah di set pada Cisco ASAv.

Gambar 22. Mengakses *WebServer* dari *Client* yang berada pada *outside zone* melalui ip yang sudah di set.

Setelah dilakukannya konfigurasi lebih lanjut, akhirnya *client* dapat mengakses *web server* melalui ip yang sudah di set pada *interface outside* Cisco ASAv, ip tersebut disesuaikan dengan subnet *interface outside* Cisco ASAV yaitu 192.168.17.85, lalu *client* yang berada di *outside* langsung dapat terhubung dengan web *server* itu dan dikarenakan sebelumnya sudah menambahkan *virtual host* dan *domain name* untuk *Web Server*, maka *web server* kemudian akan otomatis *redirect link* menjadi https dengan *domain name* jess17.net .

8. *WebClient* yang berasal dari *outside zone* mengakses Seafile Server yang berada pada *demilitarized zone* melalui *public* ip yang sudah di set pada Cisco ASAv.

Gambar 23. Mengakses *WebServer* dari *client* yang berada pada *outside zone* melalui ip yang sudah di set

Setelah langkah-langkah diatas selesai dilakukan maka Seafile Server sudah dapat diakses dari *client* yang berada di *outside zone* dengan mengkonfigurasikan ip nya sehingga *server* mempunyai seperti "*public* ip" yang dapat diakses oleh *host* apa saja yang terletak di *outside zone*, dan untuk mengaksesnya, *client* hanya perlu memasukkan ip yang sudah di set beserta *port* nya yaitu

192.168.17.91:8000, hal ini dapat dilakukan karna telah dilakukan *port forwarding* agar *client* dapat memilih *port* mana yang dapat digunakan yaitu port 8000, dimana tempat Seahub yaitu *web interface* dari *Seafile* berjalan.

4. Kesimpulan

Berdasarkan simulasi yang dibahas dalam karya ilmiah ini, dapat diambil beberapa kesimpulan yaitu sebagai berikut :

1. Simulasi jaringan dari topologi yang menggunakan aplikasi GNS3 berhasil menciptakan sistem terdistribusi Seafile dengan aksesibilitas yang baik, mudah, dan keamanan yang memadai.
2. Meskipun demikian, protokol yang digunakan pada Seafile Server tetap perlu diperhatikan lagi dan diatasi pada pengembangan selanjutnya untuk memastikan keamanan data *user* yang lebih tinggi melalui protocol HTTPS.
3. Simulasi topologi jaringan menggunakan GNS3 beserta keamanan jaringannya dengan menambahkan Cisco ASAv yang terdiri dari tiga lapisan yaitu : *INSIDE*, *OUTSIDE*, dan *DMZ*.
4. Instalasi dan pengaturan projek ini melibatkan konfigurasi *router*, NAT, *Access Control List*, Ipsec-VPN, pengujian konektivitas antar *host* lalu konektivitas setiap *host* dengan internet, dan terakhir pengaksesan *server*.

Adapun Saran untuk pengembangan selanjutnya yaitu :

1. Penerapan metode enkripsi tambahan untuk meningkatkan keamanan data.
2. Optimisasi konfigurasi pada topologi jaringan untuk dapat mengurangi terjadinya *packet loss* diantara *host* agar koneksi dapat lebih lancar.

REFERENSI

[1] Sugiyono, SISTEM KEAMANAN JARINGAN KOMPUTER MENGGUNAKAN METODE WATCHGUARD FIREBOX PADA PT GUNA KARYA INDONESIA, Jakarta: STIKOM Cipta Karya Informatika, 2016.

[2] A. N. F. D. W. Tanenbaum S, Computer Networks 6th Edition, Chicago: Pearson Higher Ed, 2021.

[3] A. N. Hidasaputra, MENGENAL KONSEP GATEWAY DAN NAT (NETWORK ADDRESS TRANSLATION), Bandar Lampung: Universitas Mitra Indonesia, 2020.

[4] Farhat, Jaringan Komputer "Application Layer", Jakarta: Universitas Gunadarma, 2019.

[5] A. P. P. F. I. S. Sutarti, IMPLEMENTASI IDS (INTRUSION DETECTION SYSTEM) PADA SISTEM KEAMANAN JARINGAN SMAN 1 CIKEUSAL, Serang: SMAN 1 CIKEUSAL, 2018.

[6] M. Ibrahim, Security comparison of ownCloud, Nextcloud, and Seafile in open source cloud storage solutions, Turku: UNIVERSITY OF TURKU, 2022.

[7] Y. A. N. Q. Y. Hu, Measurement, Analysis and Performance Improvement of the Apache Web Server, Kingston: University of Rhode Island, 1997.

[8] J. Zhang, H. Wang dan X. Xu, "Comparative study on cloud storage platforms: Seafile vs ownCloud," *Journal of Cloud Computing*, Vol. %1 dari %2vol. 6, no. 1, pp. pp. 23-30, 2017.

[9] Y. Li, K. Chen dan Q. Liu, "Secure and efficient file sharing in cloud storage," *IEEE Transactions on Cloud Computing*, vol. 6, no. 2, pp. 318-330, 2018.

[10] W. S, K. dan L. Q, "Network simulation tools:GNS3 and its applications," *International Journal of Network Management*, vol. 29, no. 3, p. e2067, 2019.