

Implementasi Perancangan Desain Transitif VPC Menggunakan Mikrotik Sebagai Routing Instance Pada Prototype Arsitektur Hub-and-Spoke GCP

Panji Yudha Tama¹, Imam Suharjo²

^{1,2}Fakultas Teknologi Informasi, Universitas Mercu Buana, Yogyakarta
e-mail: *1yudha.mawon24@gmail.com, 2imam@mercubuana-yogya.ac.id

Abstract - Virtual Private Cloud (VPC) or Virtual Private Connect (VPC) is a virtual network used in cloud computing like Google Cloud Platform (GCP). VPC enables the construction of virtual network architectures to connect services or instances within Google Cloud. One of the drawbacks of VPC on Google Cloud is its inability to forward traffic from one VPC to another (non-transitive). This shortcoming hinders the implementation of a hub-and-spoke architecture that can be implemented to facilitate the management of interconnection on multiple VPCs. Therefore, this study was conducted to investigate whether implementing a transitive VPC design using MikroTik as a routing instance in a hub-and-spoke architecture prototype can be one way to make VPCs on GCP transitive. The creation of a transitive VPC is done using Mikrotik RouterOS as a routing instance. Mikrotik is used as a router device so that MikroTik will act as a gateway for traffic between VPCs. Mikrotik is chosen because of the complete features it offers, its reliability in carrying out routing functions, and its affordable license price. The design of using Mikrotik as a routing instance successfully makes the Hub VPC transitive, so that the Hub VPC can forward traffic between spoke VPCs.

Keywords– Hub-and-Spoke, Networking, Cloud Computing, Mikrotik, Google Cloud Platform, GCP, Cloud Networking.

Abstrak – VPC atau Virtual Private Connect merupakan jaringan virtual yang digunakan dalam cloud computing seperti Google Cloud Platform (GCP). VPC memungkinkan pembangunan arsitektur jaringan virtual untuk menghubungkan service atau instance yang terdapat di Google Cloud. Salah satu kekurangan dari VPC pada Google Cloud adalah ketidakmampuannya untuk meneruskan traffic dari satu VPC ke VPC lain (non transitif). Kekurangan ini mengakibatkan terhambatnya penerapan arsitektur hub-and-spoke yang dapat diimplementasikan untuk memudahkan pengelolaan interkoneksi pada multiple VPC. Oleh karena itu, penelitian ini dilakukan untuk menjawab apakah implementasi perancangan desain transitif VPC dengan menggunakan mikrotik sebagai routing instance pada prototyping arsitektur hub and spoke dapat menjadi salah satu cara untuk membuat VPC pada GCP bersifat transitif. Pembuatan transitif VPC dilakukan dengan menggunakan Mikrotik RouterOS sebagai routing instance. Mikrotik digunakan sebagai perangkat router sehingga MikroTik akan bertindak sebagai gateway untuk traffic antar VPC. Mikrotik dipilih karena lengkapnya fitur yang ditawarkan, keandalannya dalam menjalankan fungsi routing, serta terjangkau harga license. Rancangan penggunaan Mikrotik sebagai routing instance ini berhasil membuat Hub VPC bersifat transitif, sehingga hub VPC dapat meneruskan traffic antar spoke VPC.

Kata Kunci – Hub-and-Spoke, Networking, Cloud Computing, Mikrotik, Google Cloud Platform, GCP, Cloud Networking.

I. PENDAHULUAN

VPC atau Virtual Private Connect merupakan sebuah mekanisme untuk membangun sebuah jaringan private pada layanan cloud computing seperti Google Cloud Platform [1]. Layanan cloud computing sendiri adalah layanan yang menyediakan sumber daya komputerisasi berbasis internet [2]. Sumber daya yang disediakan oleh layanan cloud computing ini nantinya dapat diakses dan dikonfigurasi dari jarak jauh menggunakan jaringan internet [3]. Layanan cloud computing saat ini menjadi layanan yang sangat diperlukan dalam lanskap perkembangan pada bidang teknologi informasi [4]

Pentingnya cloud computing dalam perkembangan teknologi informasi tidak lepas dari kebutuhan reliabilitas dan skalabilitas yang sangat tinggi di bidang teknologi dan informasi. Reliable merupakan kemampuan sumber daya komputer untuk dapat terus beroperasi dan menjalankan fungsinya [5]. Sedangkan scalable sendiri adalah bagaimana sumber daya komputer ini dapat ditingkatkan ataupun dikurangi sesuai kebutuhan pengguna untuk mendukung performa layanan [6].

Terdapat salah satu kekurangan pada VPC di layanan google cloud, kekurangan ini adalah tidak didukungnya transitif VPC [7]. Transitif VPC sendiri adalah kemampuan untuk meneruskan traffic dari satu VPC ke VPC lain. Dengan tidak hadirnya fitur transitif VPC ini mengakibatkan VPC di GCP harus selalu terhubung langsung dengan VPC lainnya bila dibutuhkan interkoneksi antar VPC. Dengan kata lain VPC ini harus saling terhubung secara mesh apabila dibutuhkan interkoneksi atau komunikasi antar VPC. Mesh sendiri merupakan arsitektur jaringan yang mengharuskan seluruh node pada jaringan terhubung satu sama lain atau fully connected [8]. Solusi selain menggunakan arsitektur mesh adalah dengan menggunakan External IP dan melakukan komunikasi antar VPC melalui jaringan internet [9]. Namun komunikasi melalui jaringan internet ini akan memberikan tambahan biaya yang cukup signifikan dari traffic egress [10].

Padahal dalam sebuah project yang kompleks biasanya dibutuhkan separasi workload VPC antar kelompok servernya. Separasi ini sering dilakukan dengan berbagai macam pertimbangan, seperti : pemisahan billing tagihan, kemudahan pengelolaan akses dari satu kelompok server ke kelompok server lain, pertimbangan skalabilitas, pemisahan. Sehingga penerapan arsitektur hub-and-spoke, atau dalam jaringan komputer biasa disebut dengan topologi star, sangat dibutuhkan untuk membangun jaringan yang saling terhubung sekaligus mudah dikonfigurasi dan dikontrol. Jika dalam topologi star di jaringan komputer pada umumnya terdapat satu buah perangkat router atau switch sebagai titik sentral dari jaringan [11], maka dalam arsitektur hub-and-spoke dalam GCP ini akan terdapat satu buah VPC yang bertindak sebagai titik sentral arsitektur yang disebut sebagai hub VPC. Sedangkan VPC lain yang terhubung ke hub VPC disebut sebagai service atau spoke VPC, yang mendapat akses route dari VPC hub [12]. Dengan sentralisasi interkoneksi antar VPC pada hub ini maka transitif VPC menjadi suatu fitur yang sangat diperlukan dalam pembangunan arsitektur hub-and-spoke.

Berdasarkan permasalahan yang diuraikan seperti diatas penulis membuat sebuah penelitian mengenai “Implementasi Perancangan Desain Transitif VPC Menggunakan Mikrotik Sebagai Routing Instance Pada Prototype Arsitektur Hub-and-Spoke GCP”. Tujuan dari penelitian ini adalah untuk membuktikan apakah penambahan Mikrotik sebagai routing instance dapat menjadi solusi untuk membuat VPC pada arsitektur hub-and-spoke dapat bersifat transitif.

II. PENELITIAN YANG TERKAIT

Dari penelitian-penelitian sebelumnya terdapat beberapa penelitian terkait. Pertama adalah “Tinjauan Performa RouterOS Mikrotik dalam Jaringan Internet: Analisis Kinerja dan Kelayakan” oleh Fauzan Prasetyo Eka Putra. Dalam jurnal tersebut peneliti melakukan serangkaian pengujian untuk analisis kinerja dan kelayakan penggunaan MikroTik RouterOS dalam jaringan. Penelitian tersebut melakukan pengujian terhadap performa dari berbagai fitur utama pada MikroTik, seperti fitur routing, firewall, queue, nat, hingga marking dengan mangle. Hasil dari penelitian tersebut menyatakan bahwa MikroTik RouterOS cukup handal sebagai router dalam sebuah jaringan [13]. Dalam penelitian tersebut juga disebutkan bahwa dengan penggunaan MikroTik dapat menambah visibilitas monitoring dimana pengelola dapat melihat seberapa besar traffic yang berjalan, selain itu dengan menggunakan MikroTik pengelola juga dapat menyesuaikan konfigurasi pada jaringannya sesuai dengan kebutuhan.

Terdapat juga penelitian lain oleh Bulbul Gupta yang berjudul “A Review on Amazon Web Service (AWS), Microsoft Azure & Google Cloud Platform (GCP) Services”. Dalam penelitian tersebut peneliti mencoba membandingkan 3 penyedia layanan cloud computing terbesar yang ada saat ini. Selain itu peneliti juga mengungkapkan tentang bagaimana masifnya penggunaan layanan cloud computing pada

berbagai bidang industri. Penelitian tersebut juga memperlihatkan naiknya trend penggunaan cloud computing dari waktu ke waktu [14]. Peneliti mengungkapkan karena tingginya kebutuhan skalabilitas sumber daya komputer sebagai server, cloud computing menjadi salah satu solusi yang banyak diadopsi. Oleh karena itu penelitian tentang bagaimana penerapan arsitektur cloud computing yang efektif perlu untuk dilakukan.

Kemudian terdapat penelitian berjudul “Desain Jaringan Komputer Terintegrasi Menggunakan Arsitektur Campus LAN” yang diteliti oleh Jeckson Sidabutar. Dalam penelitian tersebut, Jeckson Sidabuntar melakukan desain ulang arsitektur jaringan pada Universitas XYZ dengan menggunakan desain arsitektur baru yaitu Campus LAN [15]. Pada dasarnya arsitektur Campus LAN yang diimplementasikan ini sesuai dengan konsep arsitektur hub-and-spoke, dimana dalam arsitektur Campus LAN tersebut data center bertindak sebagai hub, sedangkan kantin, rektorat, kampus, dan asrama bertindak sebagai spoke. Dalam penjelasannya arsitektur yang diterapkan tersebut memiliki beberapa kelebihan pada skalabilitas, fault tolerant, quality of service, serta pada bagian security. Dalam penelitian tersebut penerapan hub-and-spoke dilakukan pada jaringan network on-premise, yaitu di sebuah universitas, sedangkan pada penelitian ini hub-and-spoke diterapkan di ranah jaringan cloud.

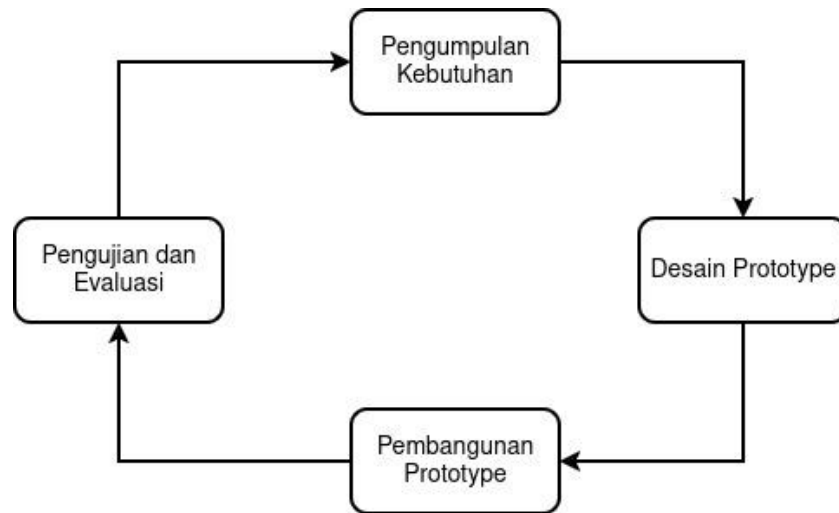
Pada layanan cloud computing selain biasa pada penggunaan resource terdapat juga beban biaya untuk penggunaan kuota jaringan, terutama untuk traffic egress atau traffic keluar dari penyedia layanan cloud computing. Dalam penelitian berjudul “Managing Cloud networking costs for data-intensive applications by provisioning dedicated network links” oleh Igor Sfiligoi dilakukan testing untuk melihat seberapa besar biaya traffic egress antara menggunakan link egress default yang disediakan oleh layanan cloud computing dengan dedicated link. Pada penelitian tersebut ditemukan bahwa penggunaan kuota dengan link default dari penyedia layanan dikenakan biaya yang sangat tinggi jika dibandingkan dengan link dedicated. Biaya yang dikenakan saat menggunakan link dari penyedia layanan adalah sebesar \$83/TB, sedangkan dengan menggunakan dedicated link biaya yang dikenakan sebesar \$42/TB, penghematan yang dilakukan hampir sebesar 50% [16]. Jika pada penelitian tersebut peneliti melakukan penghematan cost dengan dedicated link di penelitian ini cost juga dapat ditekan dengan membuat seluruh traffic antar spoke melewati infrastruktur GCP langsung, bukan melalui infrastruktur internet yang memiliki cost lebih besar.

Untuk bagian arsitektur hub-and-spoke terdapat penelitian berjudul “Hub and Spoke BGP: Leveraging multicast to improve wireless inter-domain routing” yang membahas mengenai penerapan arsitektur hub-and-spoke dalam meningkatkan inter-domain routing dalam koneksi wireless [12].

Terdapat juga penelitian berjudul “Integrasi Server On-Premise Dengan Server Cloud Menggunakan Cloud VPN dan Mikrotik IPSec Untuk Peningkatan Keamanan Koneksi” yang membahas mengenai integrasi antara server yang berada di on-premise dan cloud [17]. Perbedaan dengan penelitian ini adalah domain pembahasannya dimana penelitian sebelumnya membahas mengenai interkoneksi antara cloud dan on-premise sedangkan pada penelitian ini membahas mengenai interkoneksi dalam jaringan cloud

III. METODE PENELITIAN

Metode penelitian yang digunakan adalah metode prototyping. Metode prototyping merupakan metode dengan mengembangkan model kerja dari sebuah sistem yang kemudian akan berfungsi sebagai versi awal dari sebuah sistem [18]. Berikut adalah tahapan metode prototyping yang akan dijalankan :



Gbr. 1 Tahapan metode prototyping

A. Pengumpulan Kebutuhan

Pengumpulan kebutuhan adalah langkah untuk menentukan kebutuhan-kebutuhan atau tujuan apa saja yang perlu dicapai dalam penelitian yang dilakukan. Tujuan yang sudah ditentukan ini nantinya akan digunakan lebih lanjut untuk membangun desain.

B. Desain prototype

Desain prototype adalah langkah untuk menyusun seperti apa sistem akan dibangun, desain prototype ini dibuat untuk memenuhi kebutuhan yang sudah didefinisikan dalam pengumpulan kebutuhan.

C. Pembangunan Prototype

Setelah desain prototype selesai dijalankan langkah selanjutnya adalah membangun sistem berdasarkan desain yang telah dibuat sebelumnya.

D. Pengujian dan Evaluasi

Pada tahap pengujian dan evaluasi maka akan dilakukan uji coba pada prototype yang telah dibangun. Pengujian ini dilakukan untuk mengetahui apakah prototype yang dibangun sudah memenuhi kebutuhan atau masih ada yang perlu diperbaiki dalam prototype yang dibangun.

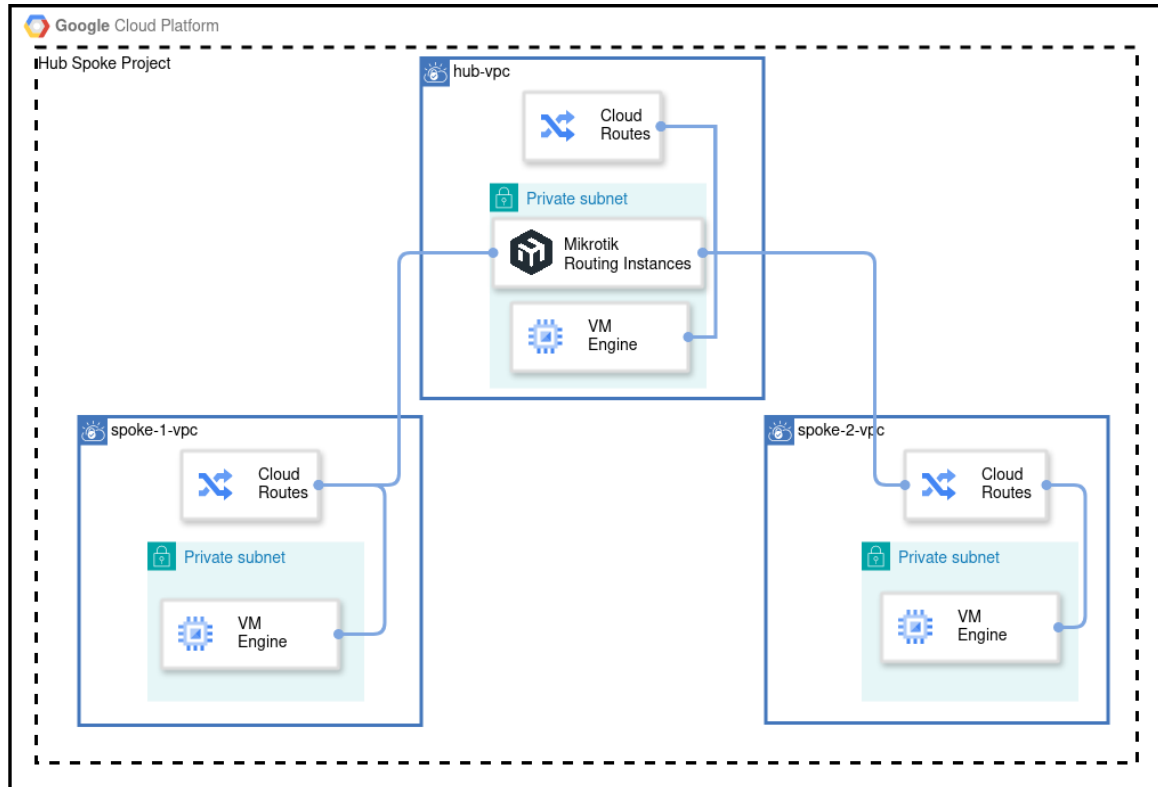
IV. HASIL DAN PEMBAHASAN

A. Pengumpulan Kebutuhan

Kebutuhan dari desain transitif VPC menggunakan mikrotik sebagai routing instance pada arsitektur hub-and-spoke GCP adalah hub-vpc dapat bersifat transitif sehingga dapat meneruskan traffic data antar spoke. Dengan kemampuan hub-vpc menjadi transitif tersebut maka diharapkan komunikasi data antar VPC dapat dilakukan dengan menggunakan IP private. IP Private sendiri merupakan IP Address yang hanya dapat digunakan di jaringan local dan bukan merupakan IP address yang dikenal di internet atau jaringan public [19]. Dengan komunikasi yang dilakukan dengan menggunakan IP private maka traffic data dilewatkan pada infrastruktur dari GCP dan tidak melewati link internet. Hal tersebut memberi dampak positif dimana tidak terdapat bottleneck pada jalur yang dilalui, selain itu komunikasi menjadi lebih secure karena data tidak melewati jaringan public, serta dengan tidak melalui internet maka diharapkan dapat menghemat cost egress dari traffic yang keluar dari GCP.

B. Desain prototype

Berdasarkan pada kebutuhan diatas maka topologi yang akan dibangun adalah sebagai berikut :



Gbr. 2 Topologi implementasi

Dalam desain diatas nantinya akan dibuat 3 buah jaringan VPC. VPC pertama bernama hub-vpc akan digunakan sebagai titik sentral hub-and-spoke yang memiliki mikrotik instance didalamnya, 2 VPC lain akan bernama spoke-1-vpc dan spoke-2-vpc yang akan berfungsi sebagai spoke yang akses ke vpc lainnya akan melalui vpc sentralnya atau dalam kasus ini adalah hub-vpc.

Pada spoke-2-vpc nantinya akan dijalankan sebuah ubuntu instances, dalam ubuntu instances tersebut akan menjalankan sebuah service http. Sedangkan pada spoke-1-vpc nantinya akan dibuat sebuah ubuntu instances yang akan berfungsi sebagai consumer dari service http yang dijalankan oleh ubuntu instances pada spoke-2-vpc.

Mikrotik akan memiliki sebuah interface main yang mengarah ke hub-vpc serta akan memiliki 2 interface tambahan yang fungsinya untuk terhubung ke spoke-1-vpc dan spoke-2-vpc.

C. Pembangunan Prototype

Dalam pembangunan prototype terdapat beberapa langkah yang harus dilakukan, dari mulai persiapan pada environment GCP, konfigurasi Mikrotik, serta konfigurasi VM Instance yang berada di spoke project.

a. Environment GCP

Diperlukan persiapan pada environment GCP yang akan digunakan, diantaranya :

- i. Create VPC pada project

Pembuatan VPC yang akan digunakan untuk membangun arsitektur hub-and-spoke.

Name ↑	Subnets	MTU ?	Mode
hub-vpc	1	1460	Custom
spoke-1-vpc	1	1460	Custom
spoke-2-vpc	1	1460	Custom

Gbr. 3 Create vpc pada google cloud.

ii. Create subnet pada hub-vpc

Pembuatan subnet untuk IP Addressing pada masing-masing VPC. Disini hub-vpc akan menggunakan segment IP 172.16.1.0/24.

Name	Region	VPC network ↑	Primary IPv4 range	Secondary IPv4 ranges
subnet-hub-vpc	asia-southeast2	hub-vpc	172.16.1.0/24	

Gbr. 4 Subnet pada Hub VPC.

iii. Create subnet pada spoke-1-vpc

Pembuatan subnet untuk spoke-1-vpc, subnet yang akan digunakan oleh spoke-1-vpc adalah segment IP 10.100.100.0/24

Name	Region	VPC network ↑	Primary IPv4 range	Secondary IPv4 ranges
spoke-1-subnet	asia-southeast2	spoke-1-vpc	10.100.100.0/24	

Gbr. 5 Subnet pada spoke-1-vpc

iv. Create subnet pada spoke-2-vpc

Pembuatan subnet untuk spoke-2-vpc, subnet yang akan digunakan oleh spoke-2-vpc adalah segment IP 10.200.200.0/24.

Name	Region	VPC network ↑	Primary IPv4 range	Secondary IPv4 ranges
spoke-2-subnet	asia-southeast2	spoke-2-vpc	10.200.200.0/24	

Gbr. 6 Subnet pada spoke-2-vpc.

b. Create VM Instance

i. Create mikrotik routing instances

Create instance Mikrotik dengan 3 interface, main interface ke hub-vpc, dan interface lain ke spoke-1 dan spoke-2

VM instances

Filter mikrotik Enter property name or value

Status	Name ↑	Zone	Recommendations	In use by	Internal IP
<input checked="" type="checkbox"/>	mikrotik-routing-instance	asia-southeast2-b			172.16.1.3 (nic0) 10.100.100.2 (nic1) 10.200.200.2 (nic2)

Gbr. 7 Mikrotik instances.

ii. Create VM Instance pada VPC spoke-1

Pembuatan VM Ubuntu sebagai consumer tester yang berada di VPC spoke-1-vpc

VM instances

Filter spoke-1-ubuntu Enter property name or value

Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP
<input checked="" type="checkbox"/>	spoke-1-ubuntu-consumer	asia-southeast2-a			10.100.100.3 (nic0)	35.219.93.124 (nic0)

Gbr. 8 Ubuntu instances VPC spoke-1.

iii. Create VM Instance pada VPC spoke-2

Pembuatan VM Ubuntu sebagai http server yang berada di VPC spoke-1-vpc

VM instances

Filter spoke-2-http-service Enter property name or value

Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP
<input checked="" type="checkbox"/>	spoke-2-http-service	asia-southeast2-a			10.200.200.3 (nic0)	35.219.4.166 (nic0)

Gbr. 9 Http service VPC spoke-2.

c. Konfigurasi Mikrotik

Terdapat dua hal dasar yang harus dikonfigurasi pada mikrotik, pertama adalah konfigurasi dhcp-client untuk pemberian IP address, selanjutnya adalah konfigurasi routing menuju subnet pada vpc spoke-1 dan spoke-2.

```
[admin@mikrotik-routing-instance] > /ip dhcp-client pr
Flags: X - disabled, I - invalid, D - dynamic
# INTERFACE USE-PEER-DNS ADD-DEFAULT-ROUTE STATUS ADDRESS
0 ether1 yes yes bound 172.16.1.3/32
1 ether2 yes yes bound 10.100.100.2/32
2 ether3 yes yes bound 10.200.200.2/32
```

Gbr. 10 DHCP client mikrotik

Mikrotik memiliki 3 interface, interface main berada di hub-vpc, sedangkan interface lain berada di spoke-1-vpc dan spoke-2-vpc. Mikrotik di GCP memiliki behavior yang sedikit unik, yaitu interface selain main akan mendapatkan network address 169.254.169.254, tampak pada gambar 10.

```
[admin@mikrotik-routing-instance] > /ip address print
Flags: X - disabled, I - invalid, D - dynamic
# ADDRESS NETWORK INTERFACE
0 D 172.16.1.3/32 172.16.1.1 ether1-hub
1 D 10.100.100.2/32 169.254.169.254 ether2-spoke-1
2 D 10.200.200.2/32 169.254.169.254 ether3-spoke-2
```

Gbr. 11 IP address mikrotik.

Setelah dhcp-client berhasil disetup perlu melakukan konfigurasi static routing agar mikrotik dapat meneruskan paket ke vpc spoke-1 dan vpc spoke-2. Karena behavior network yang sedikit berbeda seperti penjelasan diatas maka dalam konfigurasi static routing pada gateway perlu menggunakan intermediate gateway dengan format "ip-address-next-hop%interface", misal untuk ke vpc spoke-1 adalah : 169.254.169.254%ether2 [20].

```
[admin@mikrotik-routing-instance] > /ip route print
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
# DST-ADDRESS      PREF-SRC      GATEWAY      DISTANCE
0 ADS 0.0.0.0/0      172.16.1.1    1
1 A S 10.100.100.0/24 169.254.169.254... 1
2 A S 10.200.200.0/24 169.254.169.254... 1
3 ADC 169.254.169.254/32 10.100.100.2 ether2        0
4 ADS 172.16.1.0/24 172.16.1.1    1
5 ADC 172.16.1.1/32 172.16.1.3    ether1        0
```

Gbr. 12 Mikrotik static routing.

d. Konfigurasi HTTP Service

Untuk keperluan http service yang akan dibuka di ubuntu instances pada spoke-2-vpc http service akan dibuka dengan menggunakan sebuah simple app flask python dengan port 5000. Aplikasi yang dijalankan di service http tersebut adalah RBL checker. RBL sendiri adalah black list IP address karena efek dari penggunaan IP tersebut untuk spamming [21].

```
root@spoke-2-http-service:/opt/simple-webapp# systemctl start simpleapp
root@spoke-2-http-service:/opt/simple-webapp# systemctl status simpleapp
● simpleapp.service - Simple Web Server
   Loaded: loaded (/etc/systemd/system/simpleapp.service; enabled; vendor pre
   Active: active (running) since Tue 2024-06-11 10:54:18 UTC; 2 days ago
     Main PID: 2208 (python3)
        Tasks: 3 (limit: 2337)
       Memory: 18.2M
          CPU: 39.630s
     CGroup: /system.slice/simpleapp.service
            └─2208 /usr/bin/python3 /opt/simple-webapp/app.py

Jun 13 06:15:23 spoke-2-http-service python3[2208]: 161.35.44.144 - - [13/Jun/2
Jun 13 09:50:18 spoke-2-http-service python3[2208]: 162.216.150.223 - - [13/Jun
Jun 13 09:50:18 spoke-2-http-service python3[2208]: 162.216.150.223 - - [13/Jun
Jun 13 09:50:19 spoke-2-http-service python3[2208]: 162.216.150.223 - - [13/Jun
Jun 13 09:50:19 spoke-2-http-service python3[2208]: 162.216.150.223 - - [13/Jun
Jun 13 09:59:12 spoke-2-http-service python3[2208]: 198.235.24.154 - - [13/Jun/
Jun 13 13:13:08 spoke-2-http-service python3[2208]: 35.203.211.28 - - [13/Jun/2
Jun 13 13:13:08 spoke-2-http-service python3[2208]: 35.203.211.28 - - [13/Jun/2
Jun 13 13:13:09 spoke-2-http-service python3[2208]: 35.203.211.28 - - [13/Jun/2
Jun 13 13:13:09 spoke-2-http-service python3[2208]: 35.203.211.28 - - [13/Jun/2
lines 1-20/20 (END)
```

Gbr. 13 http service spoke-2-vpc

e. Konfigurasi VPC Route

VPC route digunakan untuk konfigurasi static routing untuk instances instance yang berada pada VPC tersebut. Agar antar VPC dapat berkomunikasi satu sama lain melalui mikrotik routing instance, maka VPC route perlu dikonfigurasi dengan mikrotik routing instance sebagai gateway atau next hopnya.

Select the VPC network and region for which you want to view routes

Network * Region *

Filter

Name ↑	Type	IP version	Destination IP range	Priority	Scope limits	Next hop
default-route-c0b1f7cbe3ecce23	Static	IPv4	0.0.0.0/0	1000	—	Default internet gateway
default-route-d92f5c892f52be48	Subnet	IPv4	10.100.100.0/24	0	—	Network spoke-1-vpc
spoke-1-to-spoke-2	Static	IPv4	10.200.200.0/24	1000	—	IP address 10.100.100.2 (instance mikrotik-routing-instance)

Gbr. 14 VPC Route spoke-1-vpc

Select the VPC network and region for which you want to view routes

Network *
spoke-2-vpc

Region *
asia-southeast2 (Jakarta)

VIEW REFRESH

Already displaying routes with selected parameters

Filter Enter property name or value

Name ↑	Type	IP version	Destination IP range	Priority	Scope limits	Next hop
default-route-5a092a47a1ba94dd	Static	IPv4	0.0.0.0/0	1000	–	Default internet gateway
default-route-860ca477a1bcf9a5	Subnet	IPv4	10.200.200.0/24	0	–	Network spoke-2-vpc
spoke-2-to-spoke-1	Static	IPv4	10.100.100.0/24	1000	–	IP address 10.200.200.2 (instance mikrotik-routing-instance)

Gbr. 15 VPC route spoke-2-vpc

f. Konfigurasi VPC Firewall

Secara default firewall pada VPC GCP untuk traffic ingress adalah deny, sehingga perlu konfigurasi firewall untuk allow traffic yang menuju ke mikrotik serta traffic request http yang akan masuk ke vm instances pada vpc-spoke-2.

<input type="checkbox"/>	Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Network ↑
<input type="checkbox"/>	allow-spoke-1-to-mikrotik	Ingress	mikrotik	IP ranges: 10.100.100.0/24	all	Allow	1000	spoke-1-vpc
<input type="checkbox"/>	allow-spoke-2-to-mikrotik	Ingress	mikrotik	IP ranges: 10.200.200.0/24	all	Allow	1000	spoke-2-vpc

Gbr. 16 Firewall allow traffic ke mikrotik

Filter [allow-http](#) Enter property name or value

<input type="checkbox"/>	Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Network ↑
<input type="checkbox"/>	spoke-2-vpc-allow-http	Ingress	http-server	IP ranges: 0.0.0.0/0	tcp:80, 5000	Allow	1000	spoke-2-vpc
<input type="checkbox"/>	spoke-2-vpc-allow-https	Ingress	https-server	IP ranges: 0.0.0.0/0	tcp:443	Allow	1000	spoke-2-vpc

Gbr. 17 Firewall allow traffic http ke instance spoke-2

D. Pengujian dan Evaluasi

Pada tahap pengujian dilakukan pembuktian dari transitif VPC sesuai dengan tujuan akhir dari implementasi. Pengujian dilakukan dengan melakukan testing hit koneksi dari VM Instance yang terdapat di spoke-1-vpc ke VM Instance yang terdapat di spoke-2-vpc dengan menggunakan IP Private. Proses hit ini akan dilakukan dengan curl ke service http [22], [23].

Pengujian juga akan dilakukan dengan melihat connection tracking pada router mikrotik. Fitur connection tracking pada mikrotik memungkinkan untuk melihat seluruh connection yang sedang berjalan di mikrotik [24], sehingga fitur ini dapat digunakan untuk membuktikan bahwa traffic antar spoke vpc melewati mikrotik sebagai routing instance.

Selain menggunakan curl dan connection tracking pada mikrotik, pengujian juga akan dilakukan dengan melihat hasil dari traceroute [17]. Traceroute sendiri merupakan sebuah tools yang sering digunakan di jaringan untuk melihat hop atau jalur yang dilewati sebuah paket untuk sampai ke tujuan tertentu.

Pengujian juga akan dilakukan dengan menggunakan iperf, tools ini akan digunakan untuk melihat sebesar apa kapasitas link dalam melewatkan data [25]. Sebagai catatan, selain dibatasi oleh kemampuan link iperf ini juga dibatasi oleh resource CPU dari sumber dan destinasi dalam menggenerate dan memproses data yang diterima.

Hasil dari skema pengujian yang sudah dijelaskan diatas adalah sebagai berikut :

a. Pengujian traceroute

Pada pengujian tcptraceroute dapat terlihat bahwa instance spoke-1-ubuntu-consumer yang berada di spoke-1-vpc melakukan testing tcptraceroute ke IP 10.200.200.3 yang merupakan IP dari instance

pada spoke-2-vpc yang menjalankan service http. Pada pengujian terlihat bahwa traffic berjalan melalui IP 10.100.100.2 yang merupakan IP Mikrotik routing instance. Dari pengujian ini mengindikasikan bahwa traffic bisa sampai ke spoke-2-vpc melalui hub-vpc.

```
root@spoke-1-ubuntu-consumer:/# tcptraceroute 10.200.200.3 5000
Running:
      traceroute -T -0 info -p 5000 10.200.200.3
traceroute to 10.200.200.3 (10.200.200.3), 30 hops max, 60 byte packets
 1 10.100.100.2 (10.100.100.2) 1.929 ms 1.869 ms 1.744 ms
 2 10.200.200.3 (10.200.200.3) <syn,ack> 3.564 ms 3.546 ms 3.498 ms
root@spoke-1-ubuntu-consumer:/#
```

Gbr. 18 Hasil Traceroute

b. Pengujian connection-tracking

Pengujian selanjutnya dilakukan dengan cURL dari ubuntu-consumer yang berada di spoke-1 ke http service yang dijalankan oleh ubuntu yang berada di spoke-2-vpc. Hasil pengujian menunjukkan hit cURL memberikan respon data RBL dari aplikasi yang dijalankan di spoke-2-vpc. Hal ini membuktikan bahwa spoke-1-vpc dapat berkomunikasi dengan spoke-2-vpc melalui hub-vpc.

	Src. Address	Dst. Address	Proto...	Connecti...	Timeout	TCP State	Orig./Repl. Rate	Orig./Repl. Bytes
SC	10.100.100.3:43293	10.200.200.3:5000	6 (tcp)		00:00:01	close	0 bps/0 bps	100 B/60 B
SC	10.100.100.3:45209	10.200.200.3:5000	6 (tcp)		00:00:08	close	0 bps/0 bps	100 B/60 B
SC	10.100.100.3:46979	10.200.200.3:5000	6 (tcp)		00:00:02	close	0 bps/0 bps	100 B/60 B
SC	10.100.100.3:49409	10.200.200.3:5000	6 (tcp)		00:00:03	close	0 bps/0 bps	100 B/60 B
SC	10.100.100.3:51279	10.200.200.3:5000	6 (tcp)		00:00:00	close	0 bps/0 bps	100 B/60 B
SC	10.100.100.3:51663	10.200.200.3:5000	6 (tcp)		00:00:07	close	0 bps/0 bps	100 B/60 B
SC	10.100.100.3:53355	10.200.200.3:5000	6 (tcp)		00:00:06	close	0 bps/0 bps	100 B/60 B
SC	10.100.100.3:59341	10.200.200.3:5000	6 (tcp)		00:00:04	close	0 bps/0 bps	100 B/60 B
SC	10.100.100.3:59823	10.200.200.3:5000	6 (tcp)		00:00:05	close	0 bps/0 bps	100 B/60 B
SC	10.100.100.3:60405	10.200.200.3:5000	6 (tcp)		00:00:09	close	0 bps/0 bps	100 B/60 B

Gbr. 19 Hasil test connection tracking

c. Pengujian hit cURL

Pengujian selanjutnya dilakukan dengan cURL dari ubuntu-consumer yang berada di spoke-1 ke http service yang dijalankan oleh ubuntu yang berada di spoke-2-vpc. Hasil pengujian menunjukkan hit cURL memberikan respon data RBL dari aplikasi yang dijalankan di spoke-2-vpc. Hal ini membuktikan bahwa spoke-1-vpc dapat berkomunikasi dengan spoke-2-vpc melalui hub-vpc.

```
root@spoke-1-ubuntu-consumer:/# curl "http://10.200.200.3:5000/rbl-tester/125.160.237.100" | jq
% Total % Received % Xferd Average Speed Time Time Current
Dload Upload Total Spent Left Speed
100 324 100 324 0 226 0 0:00:01 0:00:01 --:--:-- 226
{
  "BarracudaCentral": {
    "RBL_HOST": "b.barracudacentral.org",
    "RBL_STATUS": "listed",
    "RBL_WEBSITE": "http://www.barracudacentral.org/"
  },
  "RATS": {
    "RBL_HOST": "dyna.spamrats.com",
    "RBL_STATUS": "listed",
    "RBL_WEBSITE": "http://www.spamrats.com/"
  },
  "spamhaus": {
    "RBL_HOST": "zen.spamhaus.org",
    "RBL_STATUS": "listed",
    "RBL_WEBSITE": "nowebsite"
  }
}
root@spoke-1-ubuntu-consumer:/#
```

Gbr. 20 Testing hit endpoint api dengan curl

d. Pengujian bandwidth dengan iperf

Pengujian terakhir adalah test iPerf, iPerf akan menampilkan hasil berapa besar link yang dapat digunakan untuk komunikasi dari spoke-1-vpc ke spoke-2-vpc. Dari hasil pengujian iPerf menunjukkan bahwa besaran link yang dapat digunakan untuk komunikasi adalah 1 Gbps upload maupun download. Hasil 1 Gbps ini dipengaruhi juga oleh limitasi lisensi yang digunakan di MikroTik, dimana mikrotik menggunakan lisensi P1 yang memiliki batasan traffic 1 Gbps di masing-masing interfacenya.

```

root@spoke-1-ubuntu-consumer:/# iperf3 -c 10.200.200.3 -p 5201
Connecting to host 10.200.200.3, port 5201
[ 5] local 10.100.100.3 port 43184 connected to 10.200.200.3 port 5201
[ ID] Interval      Transfer      Bitrate      Retr  Cwnd
[ 5]  0.00-1.00    sec    106 MBytes    886 Mbits/sec    0   547 KBytes
[ 5]  1.00-2.00    sec    118 MBytes    991 Mbits/sec    0   547 KBytes
[ 5]  2.00-3.00    sec    115 MBytes    968 Mbits/sec    0   547 KBytes
[ 5]  3.00-4.00    sec    118 MBytes    991 Mbits/sec    0   547 KBytes
[ 5]  4.00-5.00    sec    118 MBytes    989 Mbits/sec    0   547 KBytes
[ 5]  5.00-6.00    sec    114 MBytes    958 Mbits/sec    0   547 KBytes
[ 5]  6.00-7.00    sec    110 MBytes    922 Mbits/sec    0   547 KBytes
[ 5]  7.00-8.00    sec    118 MBytes    991 Mbits/sec    0   572 KBytes
[ 5]  8.00-9.00    sec    118 MBytes    993 Mbits/sec    0   572 KBytes
[ 5]  9.00-10.00   sec    118 MBytes    988 Mbits/sec    0   572 KBytes
-----
[ ID] Interval      Transfer      Bitrate      Retr
[ 5]  0.00-10.00   sec    1.13 GBytes    968 Mbits/sec    0
[ 5]  0.00-10.04   sec    1.12 GBytes    961 Mbits/sec
iperf Done.
root@spoke-1-ubuntu-consumer:/#

```

Gbr. 21 Hasil tes iperf directional

Interface List									
Interface	Interface List	Ethernet	EoIP Tunnel	IP Tunnel	GRE Tunnel	VLAN	VRRP	Bonding	LTE
R	ether1	Ethernet							
R	ether2	Ethernet							
R	ether3	Ethernet							

	Name	Type	Actual MTU	L2 MTU	Tx	Rx	Tx Packet (p/s)	Rx Packet (p/s)
R	ether1	Ethernet	1500		23.4 kbps	3.1 kbps	3	4
R	ether2	Ethernet	1500		1007.9 kbps	988.0 Mbps	1 909	3 810
R	ether3	Ethernet	1500		1032.2 Mbps	1007.9 kbps	87 538	1 909

Gbr. 22 Traffic di interface Mikrotik

V. KESIMPULAN

Dari hasil implementasi mikrotik sebagai routing instance dalam prototyping arsitektur hub-and-spoke untuk membuat transitif vpc di GCP menghasilkan beberapa point penting: Transitif VPC pada Google Cloud dapat dilakukan dengan menambah routing instance karena secara default VPC di GCP tidak mendukung transitif. Mikrotik dapat digunakan sebagai routing instance tambahan untuk membuat transitif VPC Mikrotik dipilih karena beberapa kelebihan, seperti harga lisensi mikrotik yang terjangkau, fitur lengkap yang ditawarkan, serta performa yang stabil. Penerapan arsitektur hub-and-spoke dapat menurunkan cost traffic egress komunikasi antar spoke karena traffic tidak melalui internet. Kekurangan dari penerapan mikrotik pada uji coba diatas adalah adanya biaya tambahan untuk instances dan lisensi mikrotik. Berdasarkan beberapa point diatas maka dapat disimpulkan bahwa penambahan Mikrotik sebagai routing instance menjadi salah satu cara untuk membuat sebuah VPC yang bersifat transitif pada arsitektur hub-and-spoke di google cloud. Dari penelitian yang sudah dilakukan diatas masih terdapat use case yang perlu diteliti lebih lanjut, beberapa use case yang dapat dikembangkan untuk penelitian lebih lanjut yaitu : penerapan hub-and-spoke apabila terdapat service managed dari GCP seperti CloudSQL, perbandingan penggunaan mikrotik dengan os router lain sebagai routing instance di arsitektur hub-and-spoke.

DAFTAR PUSTAKA

- [1] S. Farizy and E. Surhayanto, "Membangun VM Windows Server Dengan Google Cloud Platform Pada PT. Zentum Intizhara," *J. Ilmu Komput. JIK*, vol. 6, no. 3, pp. 43–49, Nov. 2023.
- [2] D. Gustian, Y. Fitrissia, S. Purwatoro E.S.G.S, and W. Novayani, "Implementasi Automation Deployment pada Google Cloud Compute VM menggunakan

- Terraform,” vol. 8, no. 2, pp. 50–62, 2023, doi: 10.35314/isi.v8i1.3095.
- [3] M. R. Alamsyah, H. I. Wahanani, and M. Idhom, “Penerapan Docker Untuk Membangun Infrastruktur Private Cloud Storage Berbasis IAAS,” *J. Inform. Dan Sist. Inf.*, vol. 2, no. 2, pp. 122–131, 2021, doi: 10.2234/jifosi.v2i2.350.
- [4] D. N. Alfarizi and I. H. Iksari, “Tinjauan Literatur Terhadap Pemanfaatan Cloud Computing,” *JURIHUM J. Inov. Dan Hum.*, vol. 1, no. 1, pp. 148–154, 2023.
- [5] J. Rong-Hong, “Design of reliable networks,” *Comput. Oper. Res.*, vol. 20, no. 1, pp. 25–34, Jan. 1993, doi: 10.1016/0305-0548(93)90093-X.
- [6] E. N. Ilahi, M. Saripudin, M. A. Nugraha, G. D. A. Cardinsyah, M. F. Hikmatulloh, and M. Encep, “Mengungkap Potensi Luar Biasa dan Tantangan Menantang Cloud Computing di Era Digital,” *Karimah Tauhid*, vol. 3, no. 2, pp. 2197–2206, 2024, doi: 10.30997/karimahtauhid.v3i2.12300.
- [7] O. Lindén, “Cross region cloud redundancy: A comparison of a single-region and a multi-region approach,” Umeå University, Faculty of Science and Technology, Department of Computing Science., 2023.
- [8] J. E. A. Encarnacion and J. I. Teleron, “Innovative Advancements in Network Topologies: A Comprehensive Investigation of Mesh Network, Tree Topology, and Hypercube Network,” *Int. J. Adv. Res. Sci. Commun. Technol. IJARSCCT*, vol. 3, no. 2, pp. 394–401, 2023, doi: 10.48175/IJARSCCT-13869.
- [9] Y. Ou, “Maximizing Resource Utilization in GKE: A Case Study on IP Address Management and Budget Allocation in Google Kubernetes Engine Config Page,” *Sci. Technol. Eng. Chem. Environ. Prot.*, vol. 1, no. 1, pp. 1–4, 2023, doi: 01.0007/j.Science.2023.06.001.
- [10] E. G. M. Petrakis, V. Skevakis, P. Eliades, A. Aznavouridis, and K. Tsakos, “ModSoft-HP: Fuzzy Microservices Placement in Kubernetes,” *Electronics*, vol. 13, no. 65, pp. 1–25, 2024, doi: 10.3390/electronics13010065.
- [11] S. R. Khairyansyah, D. S. Amalia, S. Aditya, A. Sah, and D. Aribowo, “Teknologi WI-FI Menggunakan Topologi Star,” *Asos. Ris. Tek. Elektro Dan Inform. Indones.*, vol. 2, no. 2, pp. 27–33, 2022, doi: 10.61132/mars.v2i2.89.
- [12] J. Train, B. Etefia, and H. Green, “Hub and Spoke BGP: Leveraging multicast to improve wireless inter-domain routing,” *IEEE*, pp. 1–7, 2010, doi: 10.1109/AERO.2010.5446936.
- [13] F. P. E. Putra, K. Mufidah, R. M. Ilhamsyah, S. A. Efendy, and S. N. R. Barokah, “Tinjauan Performa RouterOS Mikrotik dalam Jaringan Internet: Analisis Kinerja dan Kelayakan,” *Digit. Transform. Technol. Digit.*, vol. 3, no. 2, pp. 903–910, 2023, doi: 10.47709/digitech.v3i2.3446.
- [14] T. Mufti, P. Mittal, and B. Gupta, “A Review on Amazon Web Service (AWS), Microsoft Azure & Google Cloud Platform (GCP) Services,” *Eur. Alliance Innov.*, 2021, doi: 10.4108/eai.27-2-2020.2303255.
- [15] J. Sidabutar, “Desain Jaringan Komputer Terintegrasi Menggunakan Arsitektur Campus LAN,” *J. Jaring SainTek*, vol. 2, no. 1, pp. 25–32, doi: 10.31599/jaring-saintek.v2i1.64.
- [16] I. Sfiligoi *et al.*, “Managing Cloud networking costs for data-intensive applications by provisioning dedicated network links,” *Assoc. Comput. Mach.*, no. 18, pp. 1–8, doi: 10.1145/3437359.3465563.
- [17] H. A. Al-Atsari and I. Suharjo, “Integrasi Server On-Premise Dengan Server Cloud Menggunakan Cloud VPN dan Mikrotik IPSec Untuk Peningkatan Keamanan Koneksi,” *J. Syntax Admiration*, vol. 4, no. 11, pp. 1977–1996, 2023, doi: 10.46799/jsa.v4i11.757.
- [18] M. Alda, M. H. Koto, and A. Wardani, “Implementasi Metode Prototyping Pada Rancangan Toko Tanaman Berbasis Android,” *RABIT*, vol. 8, no. 2, pp. 254–261, 2023, doi: 10.36341/rabit.v8i2.3156.
- [19] Kartika, Asran, and B. Harianto, “Utilization Of Smartphones as Human Machine Interface Display For Conveyor Control System Logo Based Web Server,” *Multidisciplinary Output Res. Actual Int. Issue MORFAI J.*, vol. 2, no. 2, pp. 255–264, 2022, doi: 10.54443/morfai.v2i2.284.
- [20] MikroTik, “Mikrotik Documentation: IP Routing,” RouterOS Documentation. Accessed: Jun. 11, 2024. [Online]. Available: <https://help.mikrotik.com/docs/display/ROS/IP+Routing>
- [21] A. Harbani and A. Sidiyantoro, “Implementasi Simple Mail Transfer Protocol Relay Pada Mail Gateway Untuk Menentukan Konten Email Spam,” *TeknoIS J. Ilm. Teknol. Inf. Dan Sains*, vol. 12, no. 1, pp. 57–66, 2022, doi: 10.36350/jbs.v12i1.130.
- [22] Rachana and R. Prasad, “Automation of Work item in ALM using cURL,” *Int. Res. J. Eng. Technol. IRJET*, vol. 7, no. 7, pp. 4955–4958, 2020.
- [23] N. Okumura, K. Ogata, and Y. Shinoda, “Formal analysis of RFC 8120 authentication protocol for HTTP under different assumptions,” *J. Inf. Secur. Appl.*, vol. 53, doi: 10.1016/j.jisa.2020.102529.
- [24] I. Muslim, F. Y. Limpraptono, and M. Ardita, “Perancangan dan Implementasi VPN Sebagai QoS Game Online Pada Jaringan Berbasis Mikrotik,” *Inst. Teknol. Nas. Malang Indones.*, 2022.
- [25] V. A. Saputro and Y. Z. Arief, “Studi Komparasi Software berbasis GUI dan CLI Terhadap Pengukuran Kualitas Throughput Jaringan Nirkabel IEEE 802.11ac,” *J. Komitika Komputasi Dan Inform.*, vol. 8, no. 1, pp. 14–21, 2024, doi: 10.31603/komitika.v8i1.11218.