



Laboratorium
Multimedia dan Internet of Things
Departemen Teknik Komputer
Institut Teknologi Sepuluh Nopember

Laporan Akhir

Praktikum Jaringan Komputer

Tunneling

Abraham Napitupulu - 5024231048

2025

Laporan Hasil Percobaan Tunneling & QoS

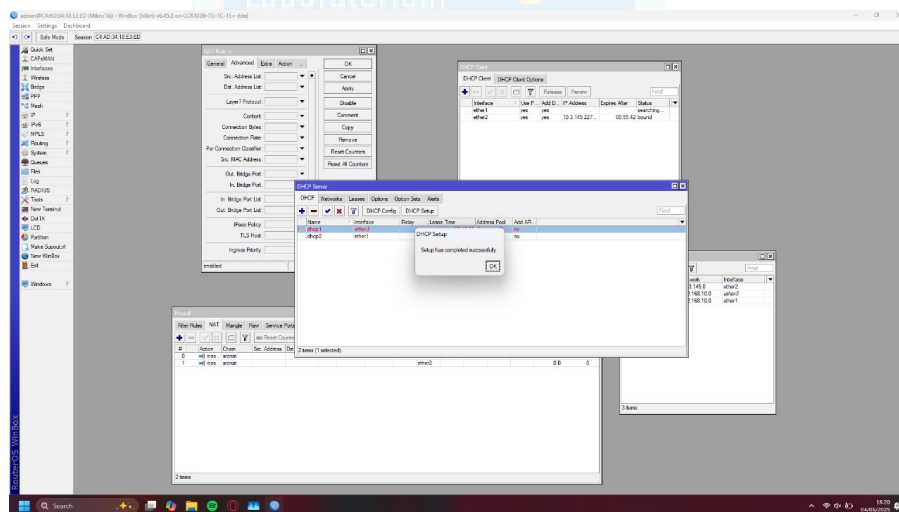
1 Langkah-Langkah Percobaan

Pada praktikum ini, saya melakukan dua skenario konfigurasi utama pada router MikroTik. Skenario pertama adalah membangun sebuah server VPN (Virtual Private Network) menggunakan protokol PPTP untuk memungkinkan akses jarak jauh yang aman. Skenario kedua adalah mengimplementasikan Quality of Service (QoS) untuk manajemen bandwidth menggunakan Simple Queue.

• Konfigurasi VPN PPTP (Remote Access)

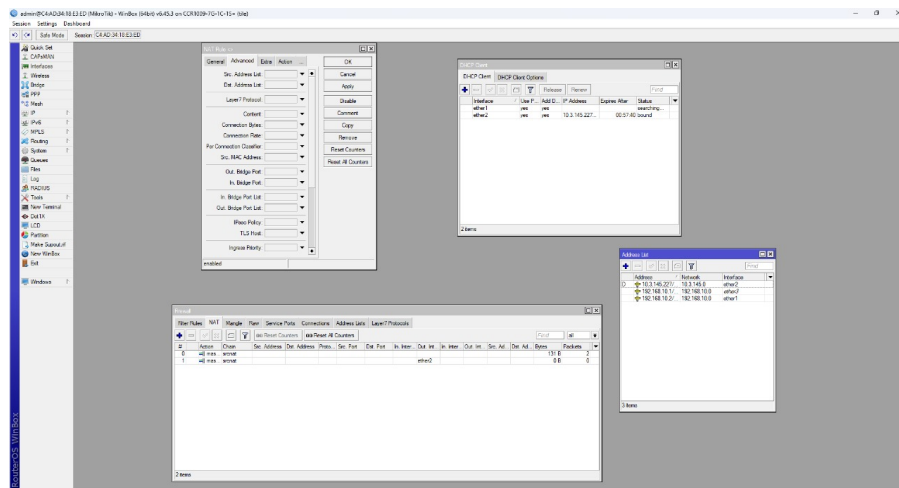
Proses ini bertujuan untuk membuat router berfungsi sebagai server PPTP, sehingga klien dari jaringan luar dapat terhubung ke jaringan lokal router.

1. **Reset dan Login Router:** Langkah awal adalah mereset router ke konfigurasi pabrik tanpa pengaturan default untuk memastikan tidak ada konflik. Setelah itu, saya login kembali menggunakan Winbox via MAC Address dengan username admin dan password kosong.
2. **Konfigurasi DHCP Client:** Saya mengkonfigurasi ether3 sebagai DHCP Client agar router mendapatkan alamat IP dan koneksi dari sumber internet (ISP). Opsi *Use Peer DNS* dan *Use Peer NTP* dipastikan aktif.



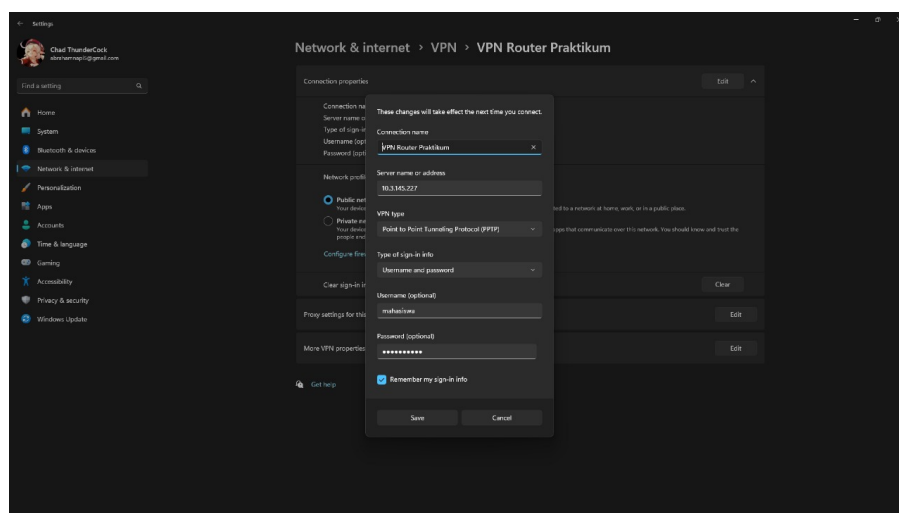
Gambar 1: Konfigurasi DHCP Client pada antarmuka ether3.

3. **Konfigurasi Firewall NAT:** Aturan NAT **masquerade** saya terapkan pada *chain* srcnat dengan *Out. Interface* mengarah ke ether3. Ini adalah langkah krusial agar semua perangkat di jaringan lokal nantinya bisa mengakses internet.
4. **Konfigurasi IP Address LAN:** Saya menambahkan alamat IP statis 192.168.10.2/24 ke antarmuka ether1 yang akan berfungsi sebagai gateway untuk jaringan lokal.



Gambar 2: Penambahan alamat IP untuk jaringan lokal pada ether1.

5. **Konfigurasi DHCP Server:** Melalui wizard *DHCP Setup*, saya mengkonfigurasi DHCP Server pada antarmuka ether1 untuk mendistribusikan IP ke klien. Sesuai modul, *Gateway for DHCP Network* diatur ke 192.168.10.2.
6. **Mengaktifkan Proxy ARP:** Pada pengaturan antarmuka ether1, mode ARP saya ubah dari *enabled* menjadi **proxy-arp** untuk membantu proses routing dan bridging pada koneksi VPN.
7. **Mengaktifkan PPTP Server:**
 - 7.a. Melalui menu *PPP*, saya membuka jendela *PPTP Server* dan mencentang kotak **Enabled**.
 - 7.b. Selanjutnya, saya membuat kredensial login pada tab *Secrets* dengan detail sebagai berikut: *Name*: mahasiswa, *Password*: praktikum123, *Service*: pptp, *Local Address*: 192.168.10.2, dan *Remote Address*: 192.168.10.5.



Gambar 3: Pembuatan user dan password untuk klien VPN pada tab Secrets.

8. **Konfigurasi PPTP Client di Windows:** Saya menambahkan koneksi VPN baru di laptop Windows, mengisikan nama koneksi, alamat server (IP publik router di ether3), tipe VPN (PPTP), serta username dan password yang telah dibuat pada langkah sebelumnya.
9. **Pengujian Koneksi VPN:** Setelah terhubung ke VPN, saya melakukan verifikasi.

- 9.a. Uji ipconfig di CMD laptop menunjukkan adanya antarmuka PPP baru dengan IP 192.168.10.5.
- 9.b. Uji ping dari laptop (klien VPN) ke gateway tunnel (192.168.10.2) berhasil.
- 9.c. Uji ping dari laptop (klien VPN) ke PC lain di jaringan lokal (ether1) juga berhasil, menandakan tunnel telah berfungsi dengan baik.

```

C:\Users\Desktop>ipconfig

IP address. . . . . : 192.168.10.5
Subnet Mask . . . . : 255.255.255.255
Default Gateway . . . . : 0.0.0.0

Wireless LAN adapter Local Area Connection* 1:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

Wireless LAN adapter Local Area Connection* 2:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

Wireless LAN adapter Wi-Fi:
Connection-specific DNS Suffix . : its.ac.id
Link-local IPv6 Address . . . . . : fe80::b0da:b0d8:fc0a:3
IP address. . . . . : 192.168.100.100
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.100.1

Ethernet adapter Bluetooth Network Connection:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

Ethernet adapter Ethernet:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

C:\Users\Desktop>ping 192.168.10.2

Pinging 192.168.10.2 with 32 bytes of data:
Reply from 192.168.10.2: bytes=32 time=1ms TTL=64
Reply from 192.168.10.2: bytes=32 time=1ms TTL=64
Reply from 192.168.10.2: bytes=32 time=1ms TTL=64
Reply from 192.168.10.2: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.10.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Desktop>

```

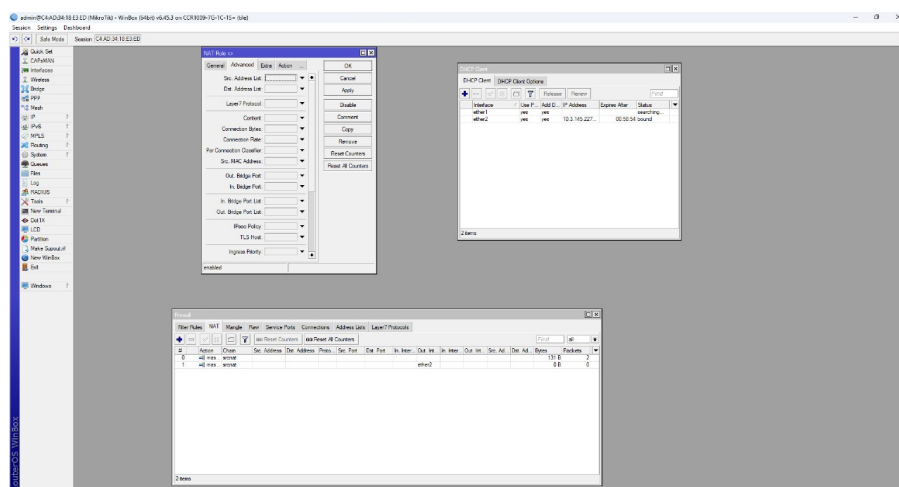
Gambar 4: Hasil uji ping yang berhasil dari klien VPN ke gateway lokal.

• Konfigurasi QoS dengan Simple Queue

Skenario ini bertujuan untuk membatasi kecepatan internet untuk klien di jaringan lokal menggunakan fitur Simple Queue. Konfigurasi ini dilakukan tanpa mereset router dari skenario sebelumnya.

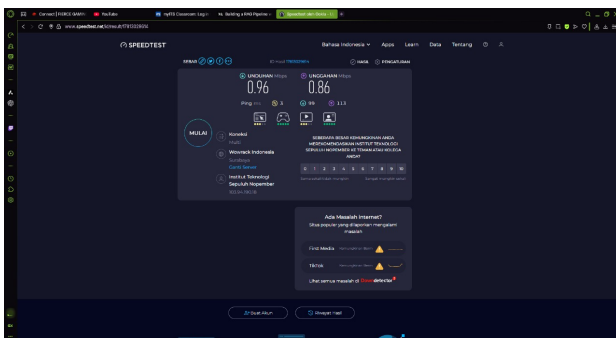
1. Membuat Aturan Simple Queue:

- 1.a. Saya mengakses menu *Queues*, lalu pada tab *Simple Queues*, saya klik ikon '+'
- 1.b. Saya menamai queue Limit-PC-Klien.
- 1.c. Pada kolom *Target*, saya memasukkan network LAN: 192.168.10.0/24.
- 1.d. *Max Limit* untuk Upload dan Download saya atur ke **1M** (1 Mbps).

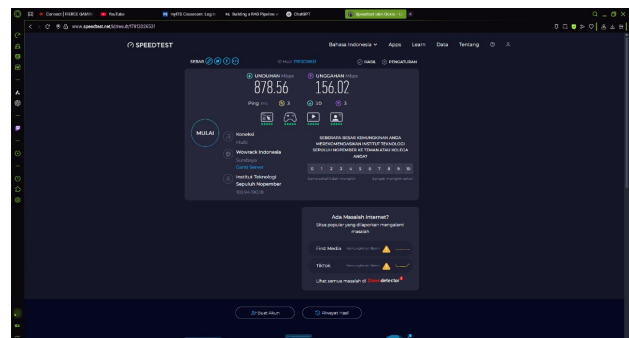


Gambar 5: Pengaturan Simple Queue untuk membatasi bandwidth jaringan lokal.

2. **Pemantauan Trafik:** Saya membuka kembali aturan queue tersebut dan pindah ke tab *Traffic* untuk melihat grafik penggunaan bandwidth secara real-time.
3. **Pengujian Efektivitas Queue:** Saya melakukan pengujian kecepatan internet dari PC klien dalam dua kondisi.
 - 3.a. **Saat Queue Tidak Aktif:** Saya menonaktifkan aturan queue dan menjalankan speed test. Hasilnya menunjukkan kecepatan maksimal yang didapat dari ISP.
 - 3.b. **Saat Queue Aktif:** Saya mengaktifkan kembali aturan queue dan menjalankan speed test lagi. Hasilnya menunjukkan kecepatan upload dan download sekarang terbatas di sekitar 1 Mbps.



Speed test saat queue nonaktif



Speed test saat queue aktif

Gambar 6: Perbandingan hasil tes kecepatan internet sebelum dan sesudah Simple Queue diaktifkan.

2 Analisis Hasil Percobaan

Berdasarkan dua skenario praktikum yang telah saya lakukan pada router MikroTik, saya dapat menganalisis fungsionalitas dan relevansi dari setiap konfigurasi yang diterapkan.

• Analisis Konfigurasi VPN PPTP

Pada skenario pertama, saya berhasil membangun sebuah server VPN untuk koneksi *remote access*. Proses ini menunjukkan beberapa poin penting. Pengaktifan **PPTP Server** pada router berfungsi sebagai "pendengar" yang siap menerima koneksi dari klien. Pembuatan kredensial pada tab **Secrets** adalah inti dari keamanan akses, di mana saya mendefinisikan username, password, serta alokasi alamat IP untuk klien yang akan terhubung. Keberhasilan ping dari laptop klien (yang mendapat IP 192.168.10.5) ke gateway router (192.168.10.2) dan ke PC lain di LAN membuktikan bahwa "terowongan" virtual telah terbentuk. Paket data dari laptop saya dibungkus menggunakan protokol PPTP, dikirim melalui internet, lalu dibuka kembali oleh router dan diteruskan ke jaringan lokal. Penggunaan **proxy-arp** pada antarmuka LAN juga krusial, karena memungkinkan router untuk "mewakili" klien VPN dalam komunikasi di jaringan lokal, sehingga seolah-olah klien VPN tersebut berada di segmen jaringan yang sama.

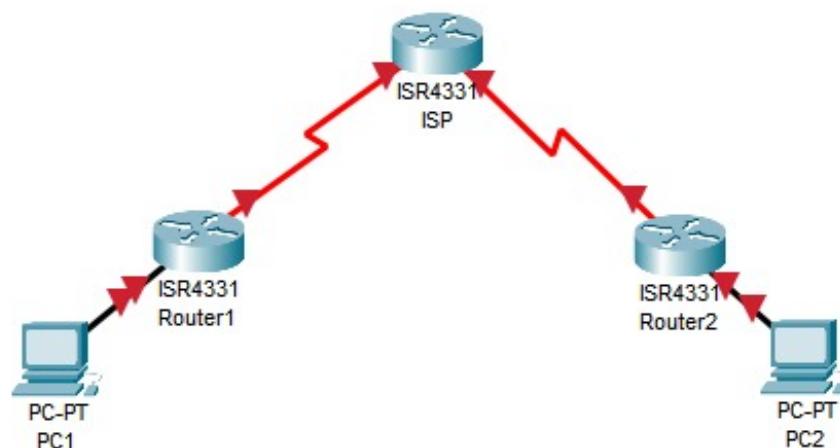
• Analisis Konfigurasi QoS dengan Simple Queue

Pada skenario kedua, saya mengimplementasikan Quality of Service (QoS) untuk membatasi bandwidth. Saya menyimpulkan bahwa **Simple Queue** adalah metode yang sangat efektif dan mudah untuk manajemen bandwidth skala kecil hingga menengah. Dengan menargetkan seluruh subnet LAN (192.168.10.0/24), saya dapat menerapkan satu aturan untuk semua klien yang terhubung. Pengaturan **Max Limit** ke 1M terbukti berhasil membatasi kecepatan unduh dan unggah klien di sekitar 1 Mbps, seperti yang ditunjukkan oleh hasil perbandingan *speed test*. Hal ini menunjukkan bahwa Simple Queue bekerja dengan cara memberlakukan batas atas (hard limit) pada lalu lintas yang cocok dengan target yang ditentukan. Fitur ini sangat relevan untuk lingkungan seperti jaringan kantor kecil atau publik di mana alokasi bandwidth yang adil dan terkontrol sangat diperlukan untuk menjaga stabilitas jaringan.

3 Tugas Modul

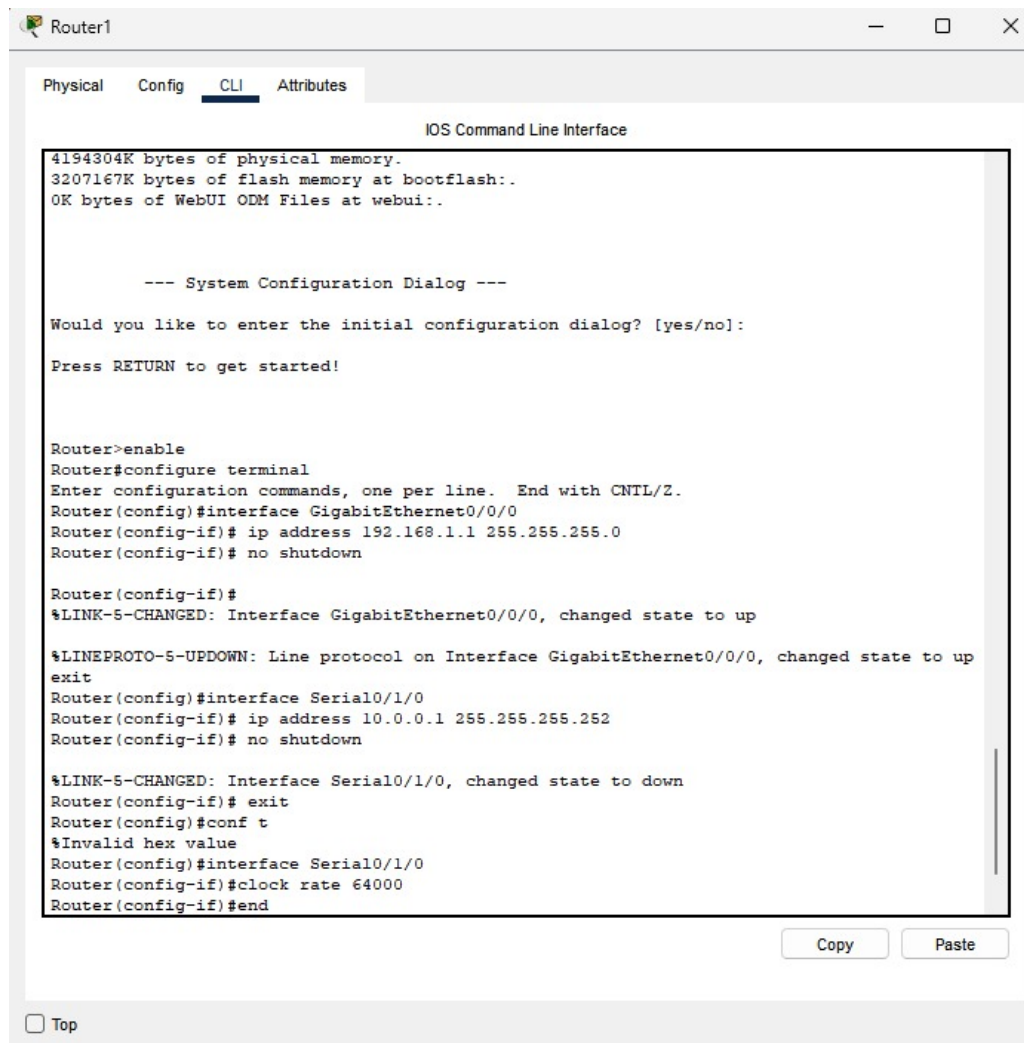
Pada bagian ini, saya melakukan simulasi untuk membangun koneksi antar dua jaringan lokal yang terpisah menggunakan *site-to-site tunnel* di Cisco Packet Tracer. Saya menggunakan GRE Tunnel sebagai metode implementasi yang paling lugas dan relevan secara konseptual.

1. **Desain Topologi dan Pengalamatan IP:** Langkah pertama adalah merancang dan membangun topologi jaringan yang terdiri dari dua situs (diwakili PC dan Router) yang terhubung melalui router perantara (ISP). Skema pengalamatan IP yang terstruktur juga saya siapkan untuk memisahkan jaringan privat, publik, dan tunnel.



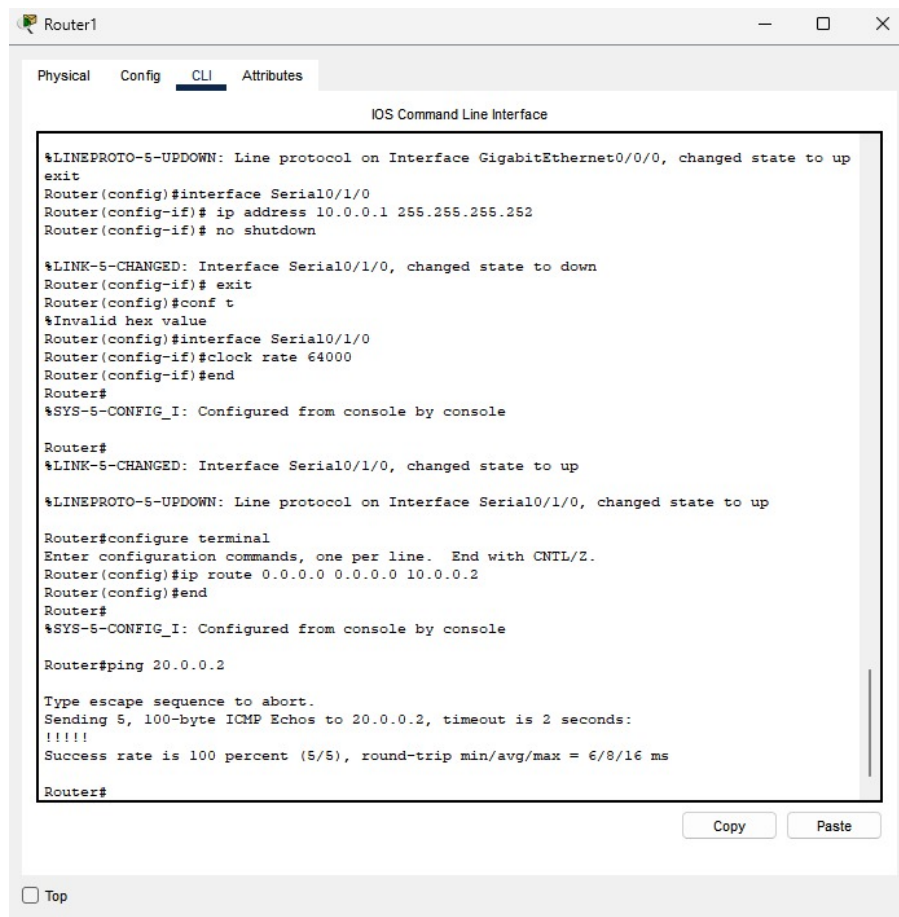
Gambar 7: Topologi jaringan site-to-site dengan 1 router ISP.

2. **Konfigurasi IP Dasar dan Perkabelan:** Saya melakukan konfigurasi IP Address pada semua perangkat, termasuk PC klien dan semua antarmuka fisik router (LAN dan Serial). Sesuai praktik terbaik yang telah diperbaiki, saya memastikan sisi DCE dari koneksi serial menerima perintah `clock rate`.



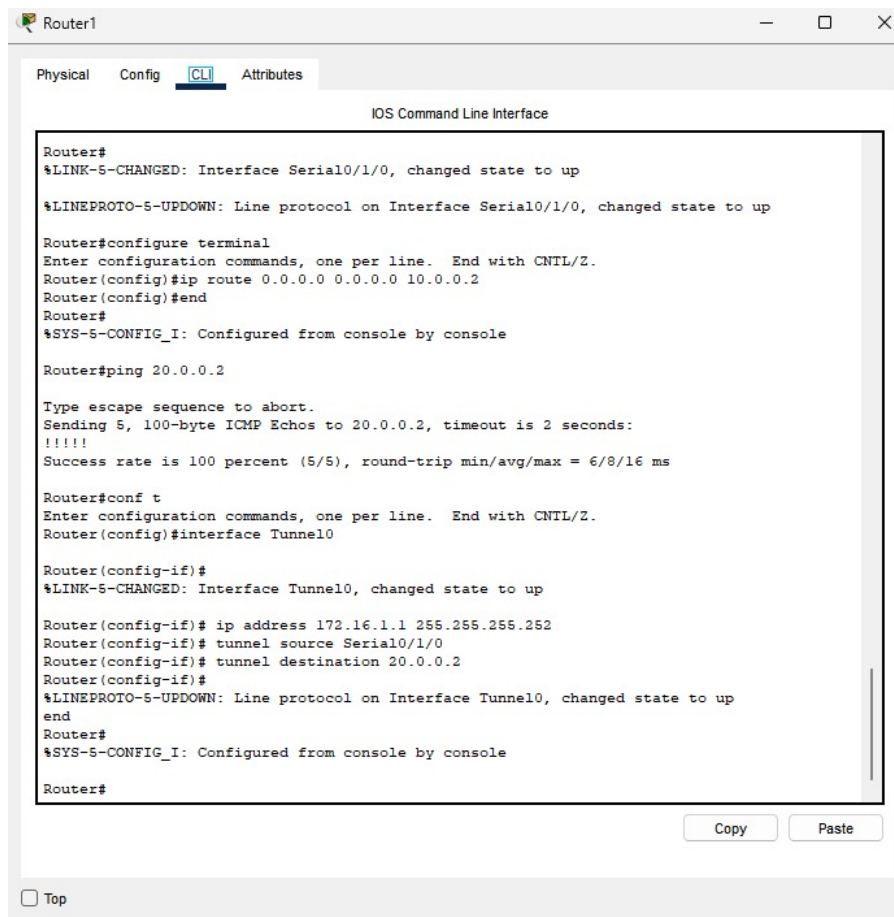
Gambar 8: Contoh konfigurasi IP Address pada salah satu router.

3. **Konfigurasi Routing Publik:** Agar router R1 dan R2 dapat berkomunikasi melalui ISP, saya menambahkan *default route* di kedua router tersebut. Rute ini mengarahkan semua lalu lintas yang tidak dikenali ke router ISP, yang secara efektif menyediakan konektivitas "internet" dasar.



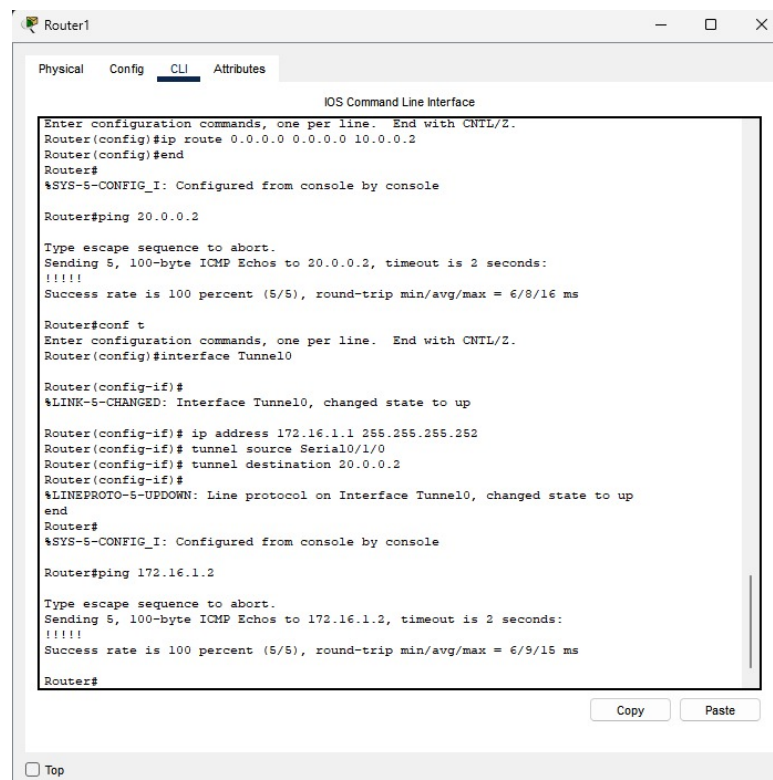
Gambar 9: Uji ping antar IP publik R1 dan R2 yang berhasil.

4. **Pembangunan GRE Tunnel:** Ini adalah inti dari simulasi. Saya membuat antarmuka virtual Tunnel0 di R1 dan R2. Konfigurasi tunnel source dan tunnel destination menggunakan alamat IP publik router, yang memerintahkan router untuk membungkus paket dan mengirimkannya melalui jaringan publik ke router tetangga.



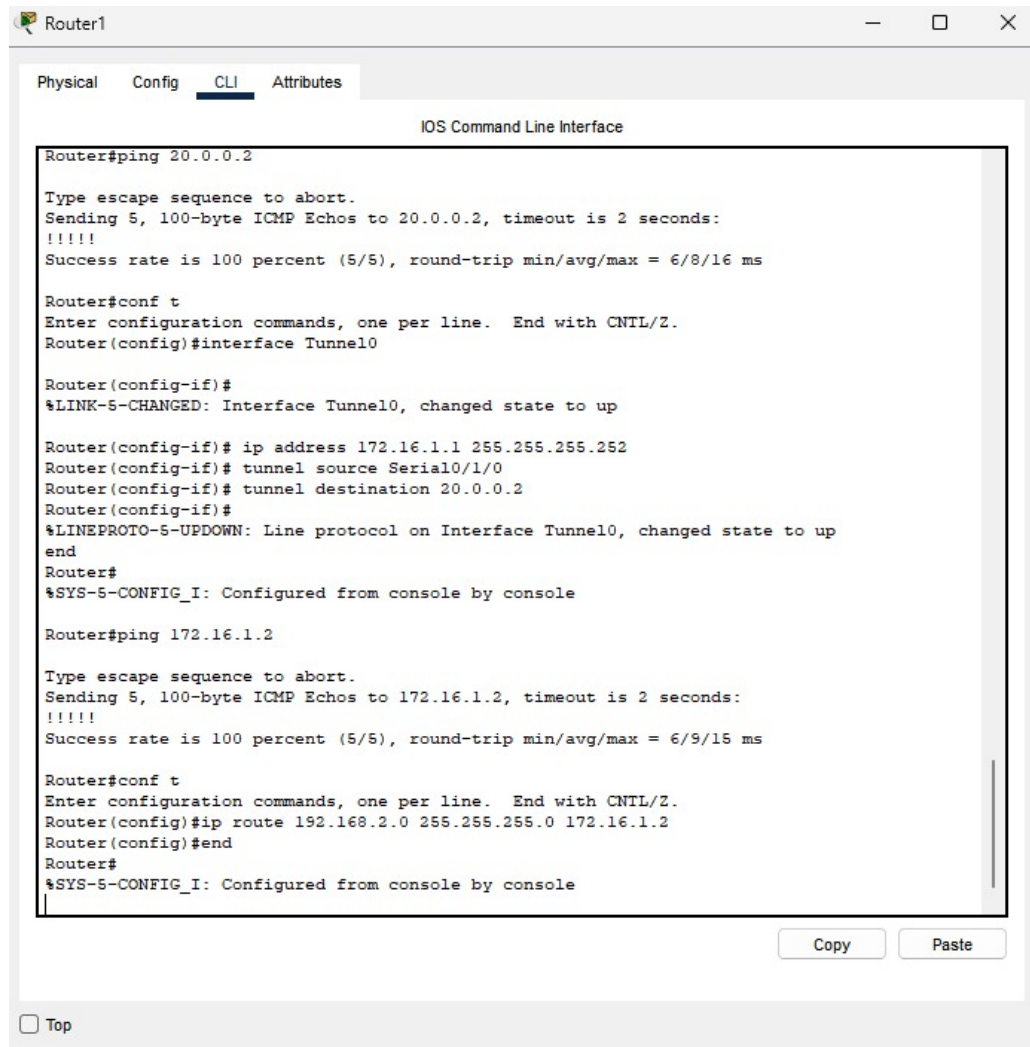
Gambar 10: Konfigurasi antarmuka Tunnel0 pada R1.

5. **Verifikasi Konektivitas Tunnel:** Saya melakukan ping ke alamat IP tunnel dari router seberang. Keberhasilan ping ini mengonfirmasi bahwa "terowongan" virtual telah berhasil terbentuk dan aktif antara R1 dan R2.



Gambar 11: Ping antar alamat IP tunnel (172.16.1.1 dan 172.16.1.2) berhasil.

6. **Routing Statis Melalui Tunnel:** Setelah tunnel aktif, saya perlu mengarahkan lalu lintas antar-LAN untuk melewatinya. Ini dilakukan dengan menambahkan rute statis di R1 untuk menjangkau LAN-2 melalui IP tunnel R2, dan sebaliknya. Tanpa rute ini, paket dari PC1 ke PC2 akan dikirim ke internet via default route, bukan melalui tunnel.



```
Router1
Physical Config CLI Attributes
IOS Command Line Interface

Router#ping 20.0.0.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 20.0.0.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 6/8/16 ms

Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface Tunnel0

Router(config-if)#
%LINK-5-CHANGED: Interface Tunnel0, changed state to up

Router(config-if)# ip address 172.16.1.1 255.255.255.252
Router(config-if)# tunnel source Serial0/1/0
Router(config-if)# tunnel destination 20.0.0.2
Router(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed state to up
end
Router#
%SYS-5-CONFIG_I: Configured from console by console

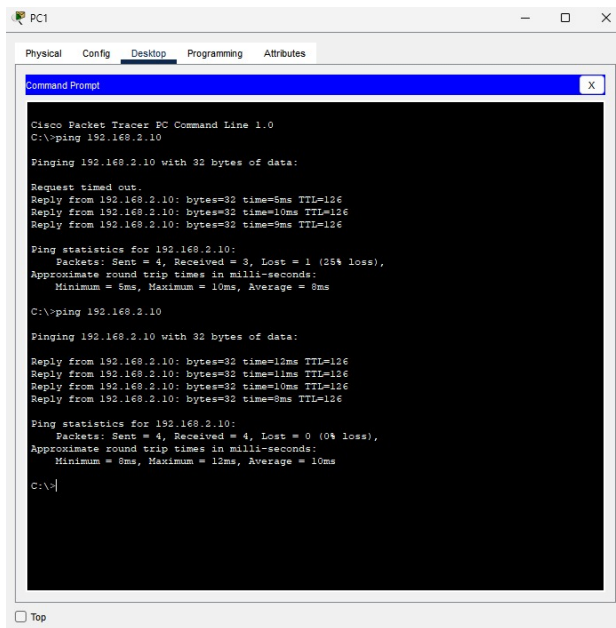
Router#ping 172.16.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 6/9/15 ms

Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 192.168.2.0 255.255.255.0 172.16.1.2
Router(config)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console
```

Gambar 12: Perintah rute statis untuk mengarahkan trafik LAN ke dalam tunnel.

7. **Pengujian Akhir dan Verifikasi Jalur:** Sebagai pembuktian akhir, saya melakukan pengujian dari PC1 ke PC2.



```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.2.10

Pinging 192.168.2.10 with 32 bytes of data:

Request timed out.
Reply from 192.168.2.10: bytes=32 time=5ms TTL=126
Reply from 192.168.2.10: bytes=32 time=10ms TTL=126
Reply from 192.168.2.10: bytes=32 time=5ms TTL=126

Ping statistics for 192.168.2.10:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 5ms, Maximum = 10ms, Average = 6ms

C:\>ping 192.168.2.10

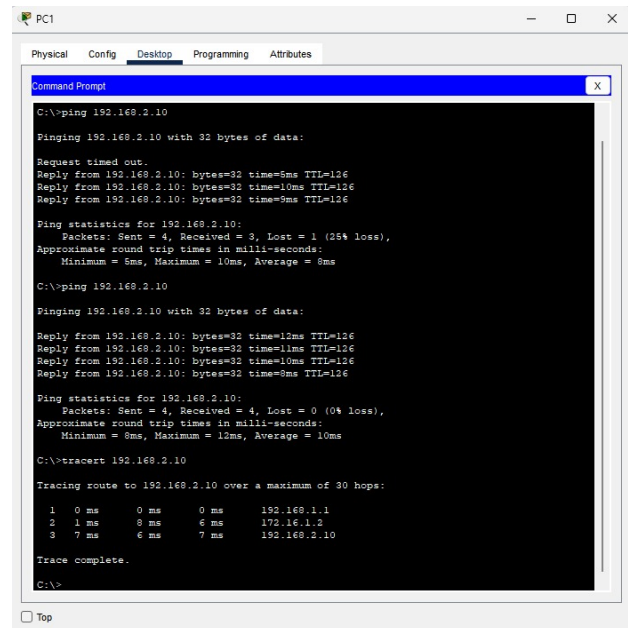
Pinging 192.168.2.10 with 32 bytes of data:

Reply from 192.168.2.10: bytes=32 time=12ms TTL=126
Reply from 192.168.2.10: bytes=32 time=11ms TTL=126
Reply from 192.168.2.10: bytes=32 time=10ms TTL=126
Reply from 192.168.2.10: bytes=32 time=9ms TTL=126

Ping statistics for 192.168.2.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 8ms, Maximum = 12ms, Average = 10ms

C:\>
```

(a) Hasil Ping Sukses



```
C:\>ping 192.168.2.10

Pinging 192.168.2.10 with 32 bytes of data:

Request timed out.
Reply from 192.168.2.10: bytes=32 time=5ms TTL=126
Reply from 192.168.2.10: bytes=32 time=10ms TTL=126
Reply from 192.168.2.10: bytes=32 time=5ms TTL=126

Ping statistics for 192.168.2.10:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 5ms, Maximum = 10ms, Average = 6ms

C:\>ping 192.168.2.10

Pinging 192.168.2.10 with 32 bytes of data:

Reply from 192.168.2.10: bytes=32 time=12ms TTL=126
Reply from 192.168.2.10: bytes=32 time=11ms TTL=126
Reply from 192.168.2.10: bytes=32 time=10ms TTL=126
Reply from 192.168.2.10: bytes=32 time=9ms TTL=126

Ping statistics for 192.168.2.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 8ms, Maximum = 12ms, Average = 10ms

C:\>tracert 192.168.2.10

Tracing route to 192.168.2.10 over a maximum of 30 hops:

  0  0 ms  0 ms  0 ms  192.168.1.1
  1  1 ms  8 ms  6 ms  192.16.1.2
  2  7 ms  6 ms  7 ms  192.168.2.10

Trace complete.

C:\>
```

(b) Hasil Traceroute

Gambar 13: Pengujian akhir menunjukkan (a) ping end-to-end berhasil dan (b) traceroute mengonfirmasi jalur melewati IP tunnel.

4 Kesimpulan

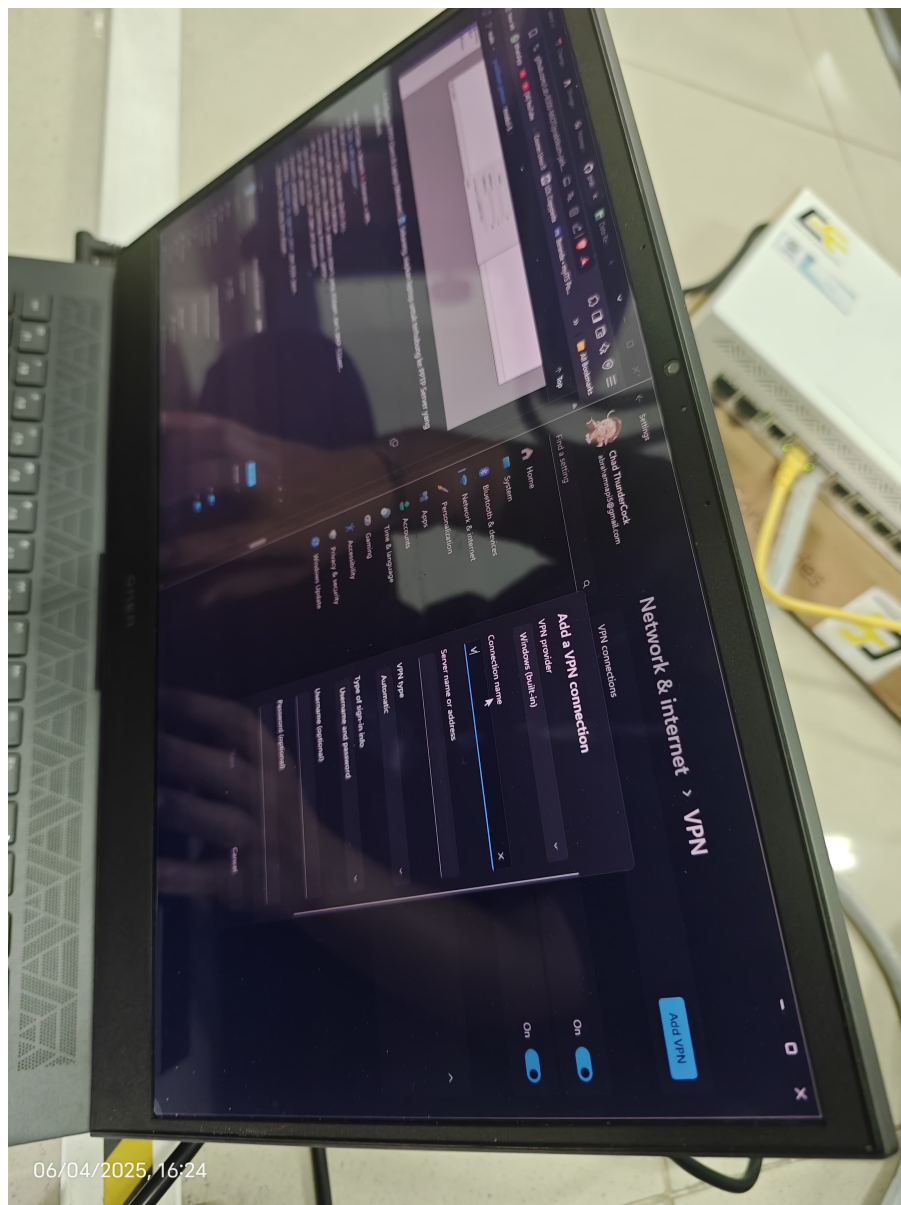
Praktikum modul kelima ini memberikan pemahaman yang utuh mengenai teknologi tunneling dan manajemen lalu lintas jaringan. Dari materi teori, saya memahami berbagai jenis protokol tunneling, dengan IPsec sebagai standar keamanan tinggi dan GRE sebagai metode enkapsulasi yang fleksibel. Pada praktikum MikroTik, saya berhasil menerapkan konsep ini secara nyata dengan membangun server VPN PPTP untuk skenario *remote access*, yang membuktikan bagaimana sebuah "terowongan" logis dapat menghubungkan klien eksternal ke jaringan internal dengan aman. Selain itu, implementasi *Simple Queue* memberikan pengalaman langsung dalam melakukan manajemen bandwidth, sebuah aspek krusial dari Quality of Service (QoS) untuk menjaga performa jaringan.

Selanjutnya, pada Tugas Modul, saya mengintegrasikan pengetahuan ini dalam skala yang berbeda melalui simulasi *site-to-site tunnel*. Penggunaan GRE tunnel di Cisco Packet Tracer secara efektif mendemonstrasikan bagaimana dua jaringan privat yang terpisah dapat disatukan melalui jaringan publik. Proses ini menegaskan pentingnya routing berlapis: routing di jaringan fisik (dasar) untuk membentuk tunnel, dan routing di atas tunnel (overlay) untuk melewati data pengguna. Secara keseluruhan, saya menyimpulkan bahwa teknologi tunneling dan QoS adalah pilar fundamental dalam rekayasa jaringan modern, yang masing-masing berfungsi untuk menyediakan konektivitas yang aman dan terukur melintasi batas-batas jaringan fisik.

5 Lampiran

Dokumentasi saat praktikum

Berikut adalah dokumentasi saat praktikum Modul 4 berlangsung.



06/04/2025, 16:24

Gambar 14: Dokumentasi praktikum.