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

Laporan Akhir
Praktikum Jaringan Komputer

Firewall & NAT

Abraham Napitupulu - 5024231048

2025

Laporan Hasil Percobaan Firewall & NAT

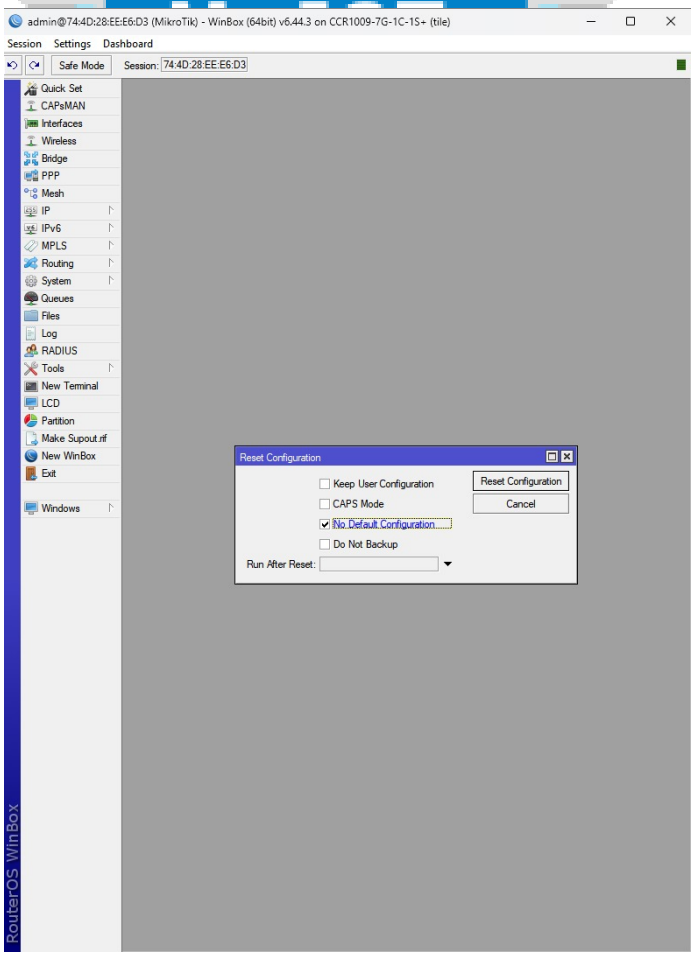
1 Langkah-Langkah Percobaan

Pada praktikum ini, dilakukan serangkaian konfigurasi pada router MikroTik untuk mengimplementasikan fungsionalitas Network Address Translation (NAT) dan Firewall. Proses ini bertujuan untuk menyediakan konektivitas internet ke jaringan lokal sekaligus mengamankannya dengan aturan filter tertentu.

• Konfigurasi Awal Router

Langkah pertama adalah memastikan router berada dalam kondisi konfigurasi default kosong untuk menghindari konflik pengaturan.

1. **Reset Konfigurasi Router:** Perangkat router direset ke pengaturan awal melalui aplikasi Winbox.
   (a) Akses menu *System -> Reset Configuration*.
   (b) Aktifkan opsi **No Default Configuration** dengan mencentangnya.
   (c) Klik tombol *Reset Configuration* untuk memulai proses.



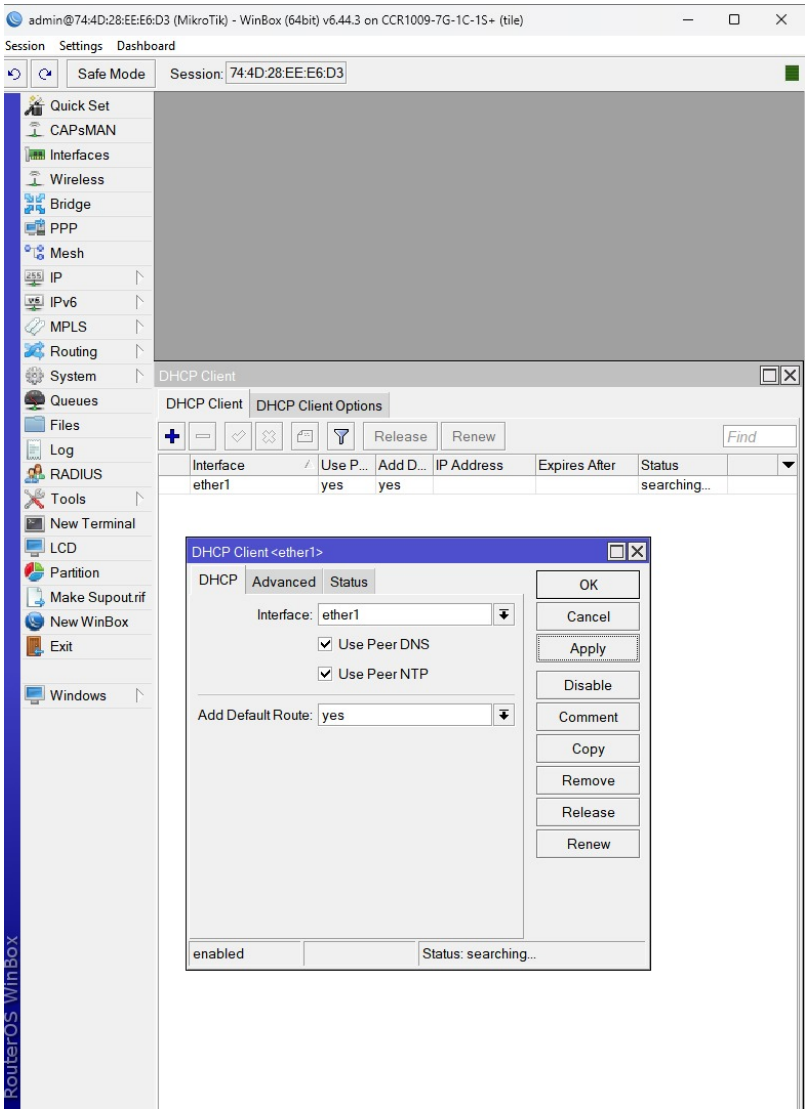
Gambar 1: Pengaturan reset router dengan opsi "No Default Configuration"

2. **Login ke Router:** Setelah router selesai melakukan reset dan restart, dilakukan proses login kembali menggunakan Winbox dengan MAC address.
   (a) Username yang digunakan adalah `admin`.
   (b) Kata sandi dikosongkan.

• Konfigurasi Jaringan dan NAT

Tahap ini meliputi pengaturan koneksi ke internet (WAN) dan distribusi IP ke jaringan lokal (LAN), serta konfigurasi NAT.

1. **Konfigurasi DHCP Client (ether1):** Antarmuka `ether1` dikonfigurasi untuk menerima alamat IP secara dinamis dari sumber internet.
   (a) Akses menu *IP* -> *DHCP Client*, lalu klik ikon '+'.
   (b) Pilih *Interface*: `ether1`.
   (c) Klik *Apply* dan pastikan status menunjukkan **bound**.



Gambar 2: Status DHCP Client menunjukkan "bound" setelah konfigurasi

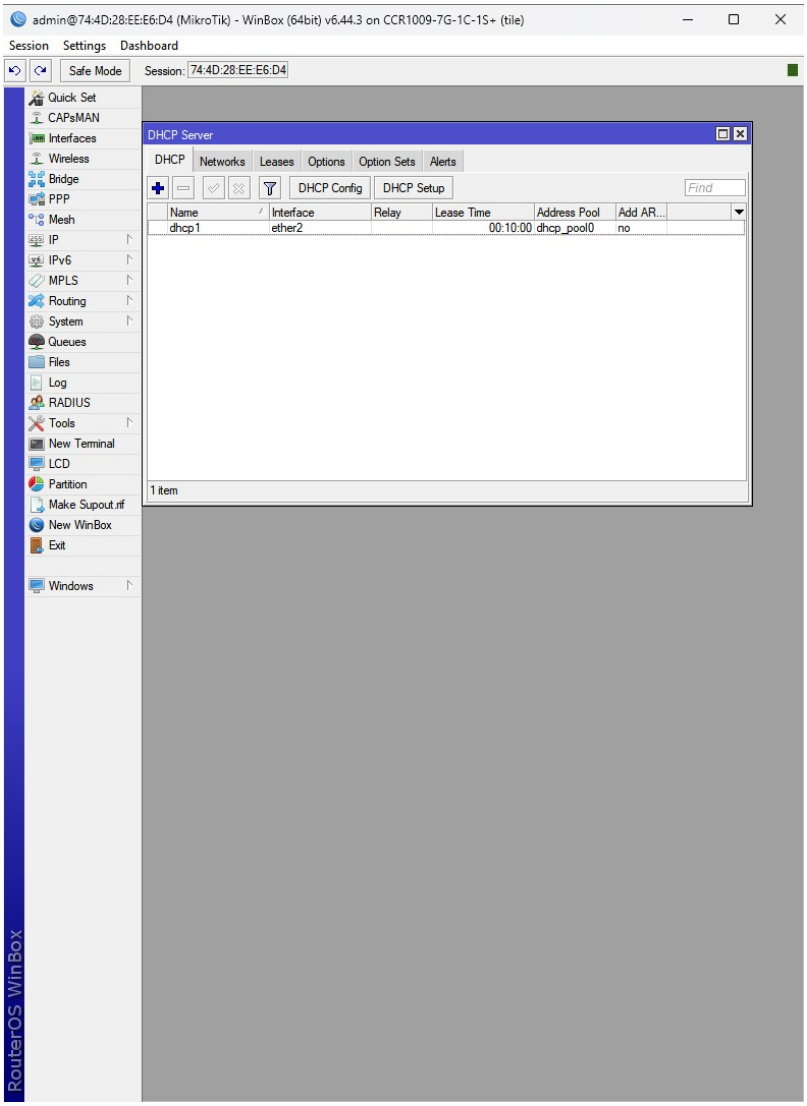
2. **Penambahan Alamat IP LAN (ether7):** Alamat IP statis ditambahkan ke antarmuka `ether7` yang akan terhubung ke jaringan lokal.

   (a) Akses menu *IP* -> *Addresses*, lalu klik ikon '+'.

   (b) Masukkan *Address*: `192.168.10.1/24`.

   (c) Pilih *Interface*: `ether7`.

3. **Konfigurasi DHCP Server:** Router diatur untuk mendistribusikan alamat IP secara otomatis ke perangkat klien di jaringan lokal melalui `ether7`. Proses ini dilakukan melalui wizard *DHCP Setup*.



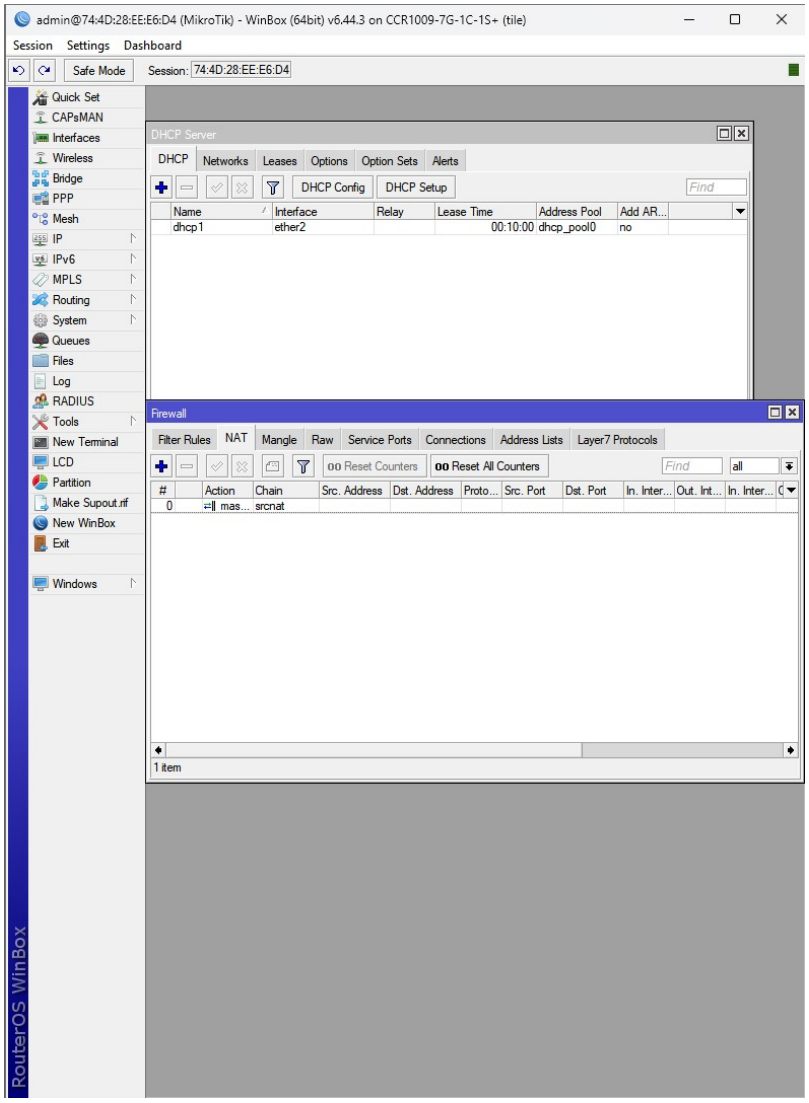
Gambar 3: Jendela wizard "DHCP Setup" untuk antarmuka ether7

4. **Konfigurasi NAT:** Aturan NAT dibuat agar semua perangkat di jaringan lokal dapat mengakses internet menggunakan satu IP publik dari router.

   (a) Akses menu *IP* -> *Firewall* -> *NAT*, lalu klik ikon ‘+‘.

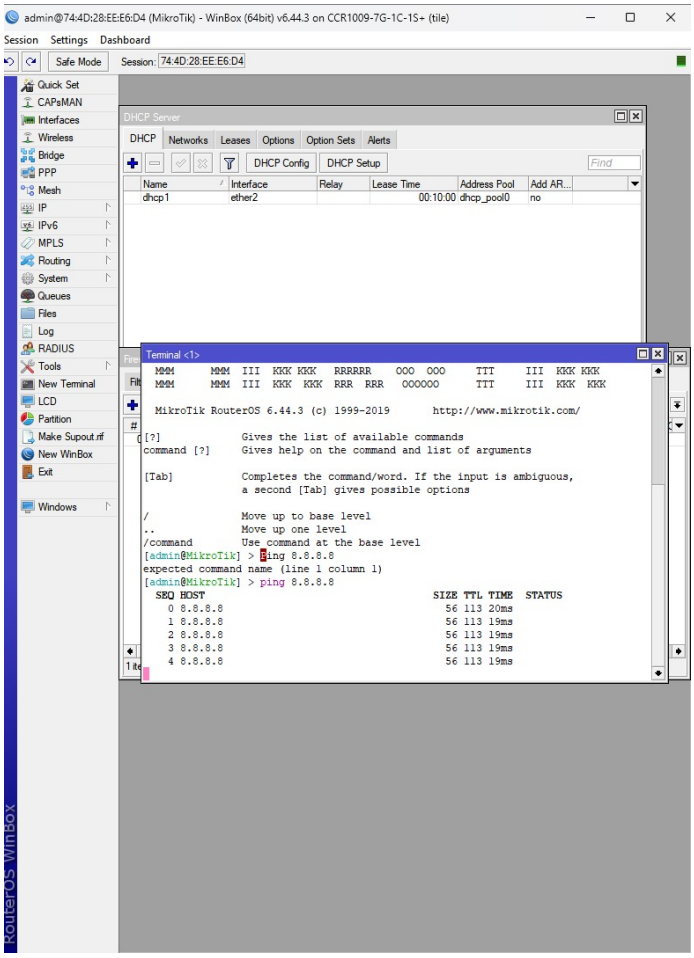
   (b) Pada tab *General*, atur *Chain*: **src-nat**.

   (c) Pada tab *Action*, atur *Action*: **masquerade**.



Gambar 4: Konfigurasi aturan NAT Masquerade pada tab Action

5. **Uji Konektivitas NAT:** Dilakukan uji `ping` dari terminal Winbox ke server Google untuk memastikan router telah terhubung ke internet.



Gambar 5: Hasil uji ping ke 8.8.8.8 yang berhasil setelah NAT dikonfigurasi

• Konfigurasi Aturan Firewall

Dua aturan filter ditambahkan pada firewall untuk membatasi jenis lalu lintas data tertentu dari jaringan lokal.

1. **Pemblokiran ICMP (Ping):** Aturan dibuat untuk memblokir semua lalu lintas ICMP yang berasal dari jaringan lokal.

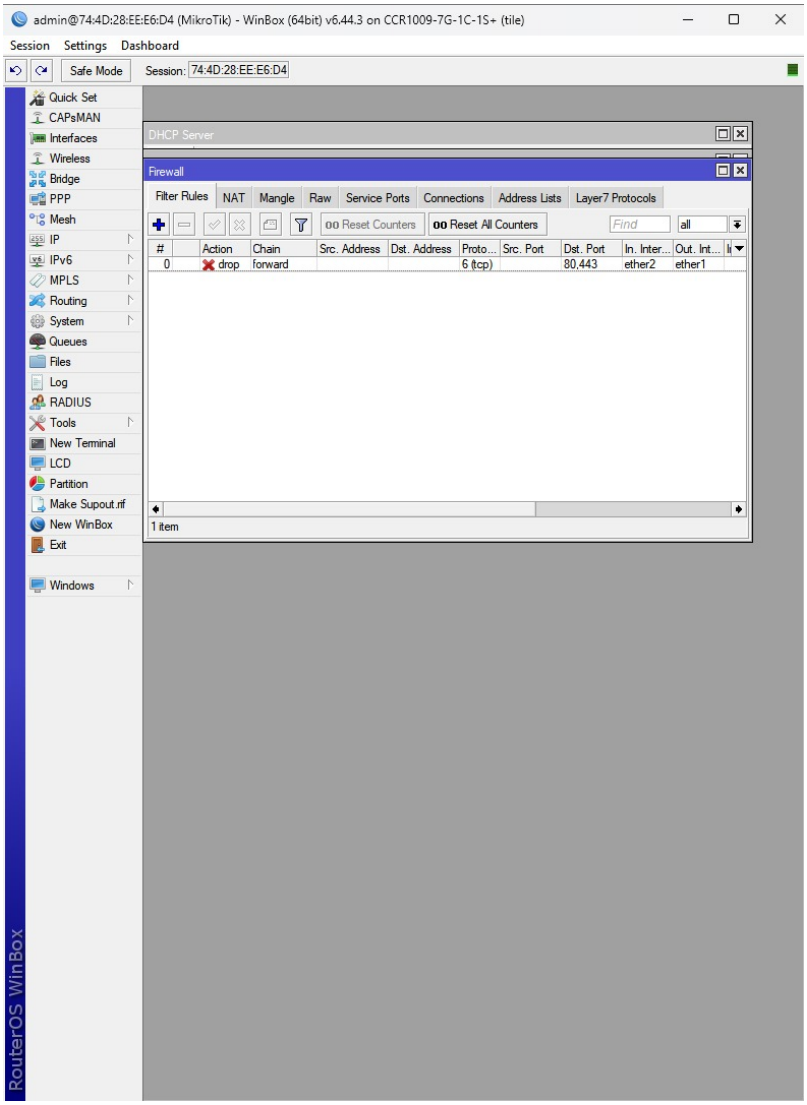
   (a) Akses menu *IP* -> *Firewall* -> *Filter Rules*, lalu klik '+'.

   (b) *Chain*: **forward**.

   (c) *Protocol*: **icmp**.

   (d) *In. Interface*: `ether7`.

   (e) *Action*: **drop**.



Gambar 6: Aturan firewall untuk melakukan "drop" pada protokol ICMP

2. **Pemblokiran Konten Website:** Aturan dibuat untuk memblokir akses ke situs web yang mengandung kata kunci "speedtest".

   (a) Buat aturan baru di *Filter Rules*.

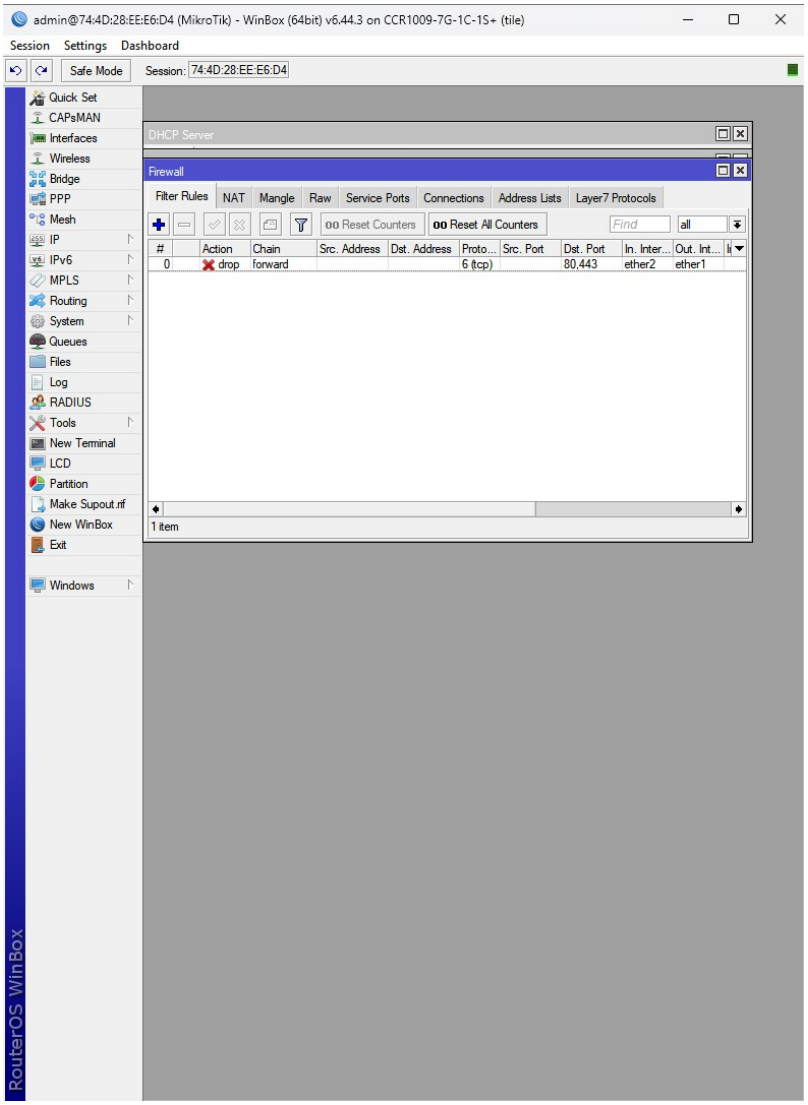
   (b) *Chain*: **forward**.

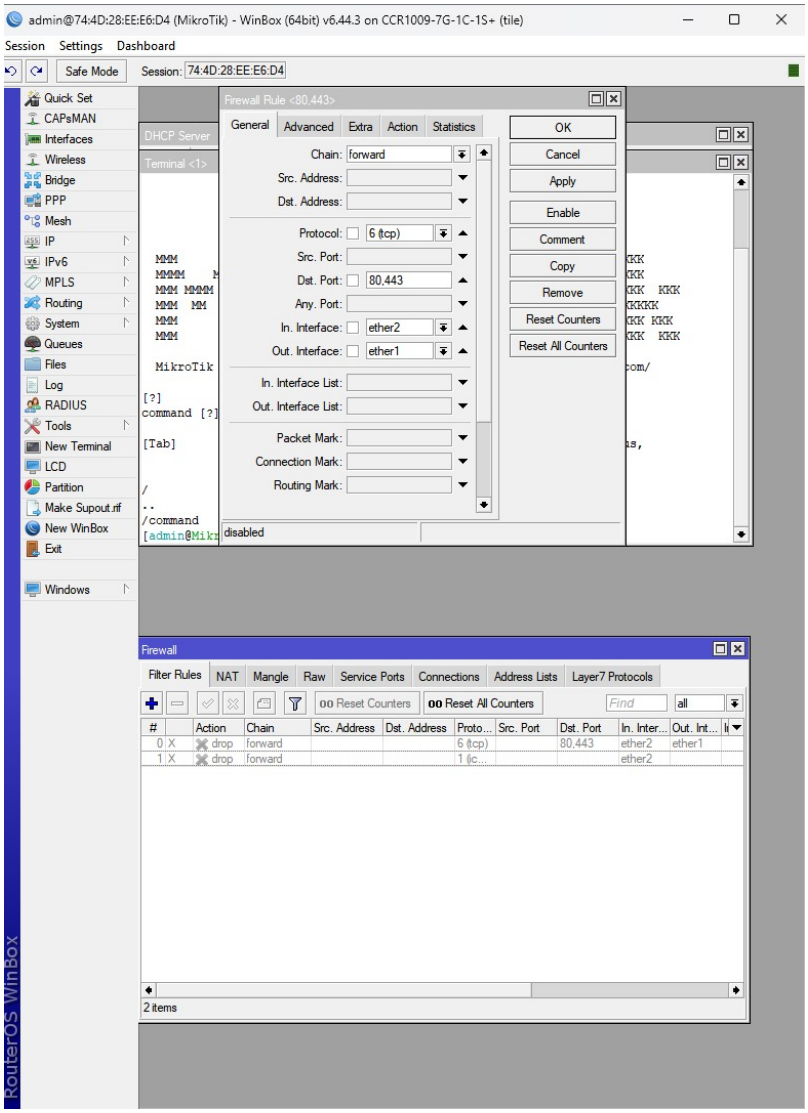
   (c) *Protocol*: **tcp**, dengan *Dst. Port*: `80,443`.

   (d) *In. Interface*: `ether7` dan *Out. Interface*: `ether1`.

   (e) Pada tab *Advanced*, atur *Content*: `speedtest`.

   (f) *Action*: **drop**.


admin@74:4D:28:EE:E6:D4 (MikroTik) - WinBox (64bit) v6.44.3 on CCR1009-7G-1C-1S+ (tile)
Session Settings Dashboard
Safe Mode
Session: 74:4D:28:EE:E6:D4
Quick Set
CAPsMAN
Interfaces
Wireless
Bridge
PPP
Mesh
IP
IPv6
MPLS
Routing
System
Queues
Files
Log
RADIUS
Tools
New Terminal
LCD
Partition
Make Supout.rif
New WinBox
Exit
Windows
RouterOS WinBox
DHCP Server
Firewall
Filter Rules
NAT
Mangle
Raw
Service Ports
Connections
Address Lists
Layer7 Protocols
Reset Counters
Reset All Counters
Find
all
Action Chain Src. Address Dst. Address Proto... Src. Port Dst. Port In. Inter... Out. Int...
0 drop forward 6 (tcp) 80,443 ether2 ether1
1 item

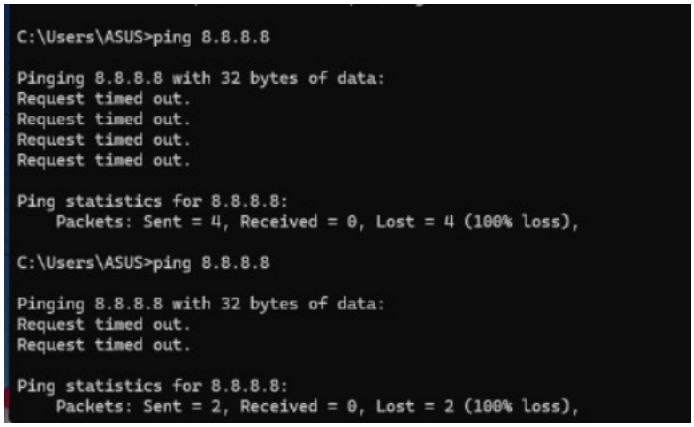


Gambar 7: Pengaturan pemblokiran konten pada tab Advanced

• Pengujian Fungsionalitas

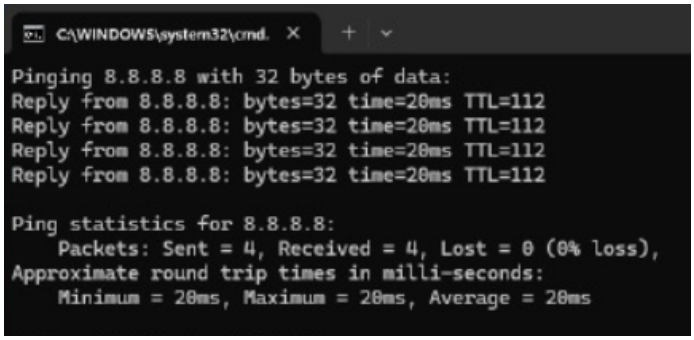
Pengujian akhir dilakukan dari sisi klien (laptop) untuk memverifikasi efektivitas aturan firewall yang telah dibuat.

1. **Pengujian Blokir ICMP:** Perintah `ping 8.8.8.8` dijalankan dari Command Prompt laptop.



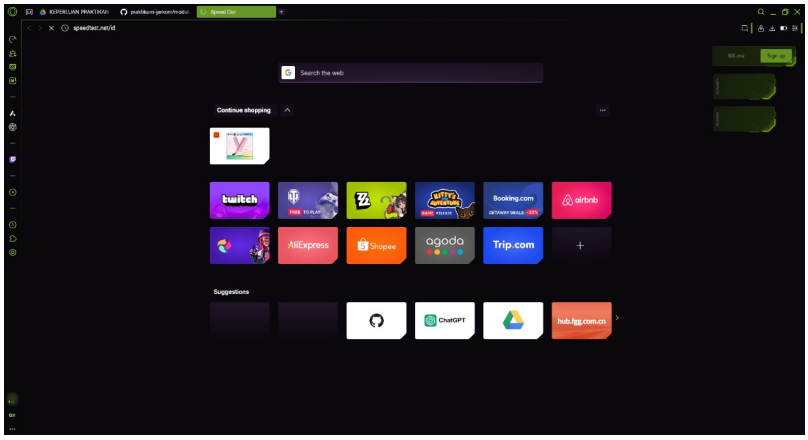
Gambar 8: Hasil uji ping menunjukkan "Request Timed Out" saat aturan firewall ICMP aktif

2. Setelah aturan firewall ICMP dinonaktifkan (*disable*), uji ping diulangi.



Gambar 9: Hasil uji ping berhasil setelah aturan firewall ICMP dinonaktifkan

3. **Pengujian Blokir Konten:** Dilakukan upaya untuk mengakses situs web `www.speedtest.net` dari peramban web.



Gambar 10: Peramban web gagal memuat situs saat aturan pemblokiran konten aktif

4. Setelah aturan pemblokiran konten dinonaktifkan, situs web yang sama dicoba diakses kembali.

2 Analisis Hasil Percobaan

Berdasarkan serangkaian langkah percobaan yang telah saya lakukan pada router MikroTik, saya melakukan analisis terhadap fungsionalitas dan hasil dari setiap konfigurasi yang diterapkan.

• Analisis Konfigurasi NAT

Pada percobaan ini, saya mengonfigurasi NAT menggunakan metode **masquerade**. Dari pengamatan saya, metode ini berhasil menyediakan konektivitas internet untuk seluruh perangkat di jaringan lokal (LAN). Keberhasilan ini disebabkan oleh cara kerja *masquerade* yang secara dinamis dan otomatis menerjemahkan alamat IP privat dari jaringan lokal (misalnya `192.168.10.1`) menjadi satu alamat IP publik yang didapat oleh router pada antarmuka `ether1`. Ketika server di internet membalas, router menggunakan tabel pelacakan koneksi (*connection tracking*) untuk mengetahui secara pasti ke perangkat privat mana paket balasan tersebut harus diteruskan. Hal ini terbukti dari uji `ping` ke `8.8.8.8` dari terminal router yang berhasil, menandakan router itu sendiri sudah terhubung ke internet dan siap meneruskan koneksi.

• Analisis Aturan Firewall ICMP

Saya membuat aturan firewall untuk memblokir protokol ICMP dengan aksi **drop**. Saat saya melakukan uji `ping` dari laptop klien, hasilnya adalah *Request Timed Out* (RTO). Saya menganalisis bahwa kegagalan ping ini disebabkan oleh aksi `drop` yang saya pilih. Aksi ini secara diam-diam membuang paket ICMP yang masuk ke router dari jaringan lokal tanpa mengirimkan notifikasi error apapun kembali ke pengirim. Akibatnya, laptop saya sebagai pengirim hanya bisa menunggu balasan yang tidak akan pernah datang hingga batas waktu habis. Ini berbeda dengan aksi `reject`, yang akan secara aktif mengirimkan balasan "Destination Unreachable". Keberhasilan uji `ping` setelah aturan ini dinonaktifkan mengonfirmasi bahwa aturan tersebut bekerja sesuai fungsinya.

• Analisis Aturan Firewall Pemblokiran Konten

Aturan pemblokiran konten yang menargetkan kata kunci `speedtest` juga berhasil saya implementasikan. Aturan ini bekerja pada Layer 7 model OSI, di mana router secara aktif memeriksa aliran data dari paket TCP yang menuju port 80 (HTTP) dan 443 (HTTPS). Ketika string `speedtest` terdeteksi dalam data (misalnya, dalam nama domain atau konten halaman), firewall akan langsung memutuskan koneksi dengan aksi `drop`. Inilah sebabnya saat saya mencoba mengakses situs `www.speedtest.net`, peramban web hanya terus memuat tanpa menampilkan konten apapun. Situs tersebut baru dapat diakses dengan normal setelah aturan filter ini saya nonaktifkan, yang membuktikan efektivitas pemfilteran berbasis konten.

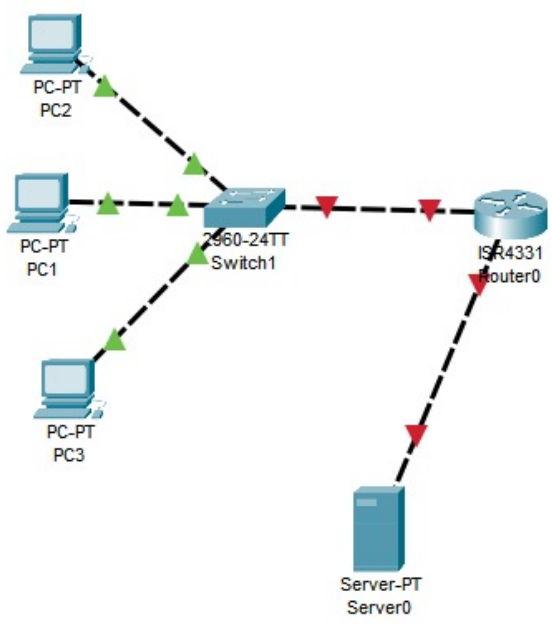
3 Tugas Modul

Pada bagian ini, saya melakukan simulasi jaringan menggunakan Cisco Packet Tracer untuk menerapkan konsep NAT dan berbagai skenario firewall menggunakan Access Control List (ACL) pada topologi yang telah ditentukan.

• Topologi dan Konfigurasi Awal

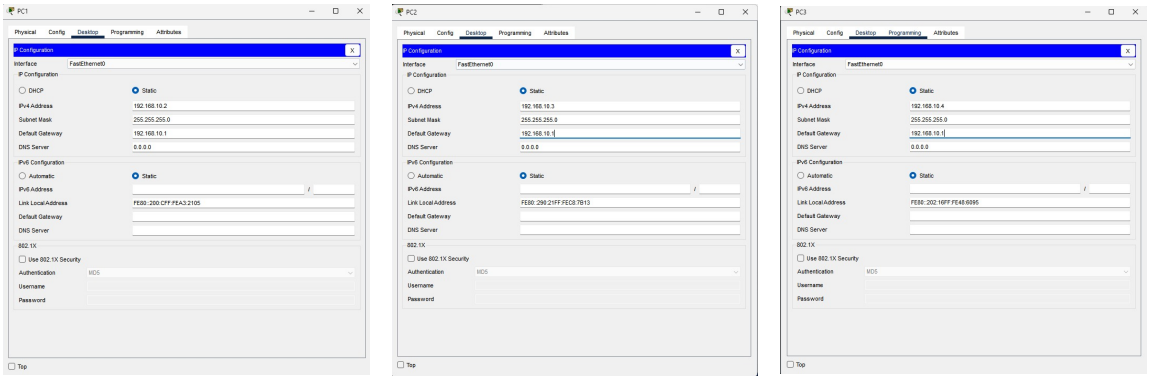
Langkah awal adalah membangun topologi jaringan dan melakukan konfigurasi IP dasar pada setiap perangkat.

1. **Pembuatan Topologi Jaringan:** Saya menempatkan 1 Router, 1 Switch, 3 PC, dan 1 Server pada workspace Cisco Packet Tracer dan menghubungkannya dengan kabel yang sesuai. Pengkabelan ini mengikuti praktik terbaik, di mana port FastEthernet pada switch digunakan untuk PC, dan port GigabitEthernet digunakan untuk koneksi uplink ke router.

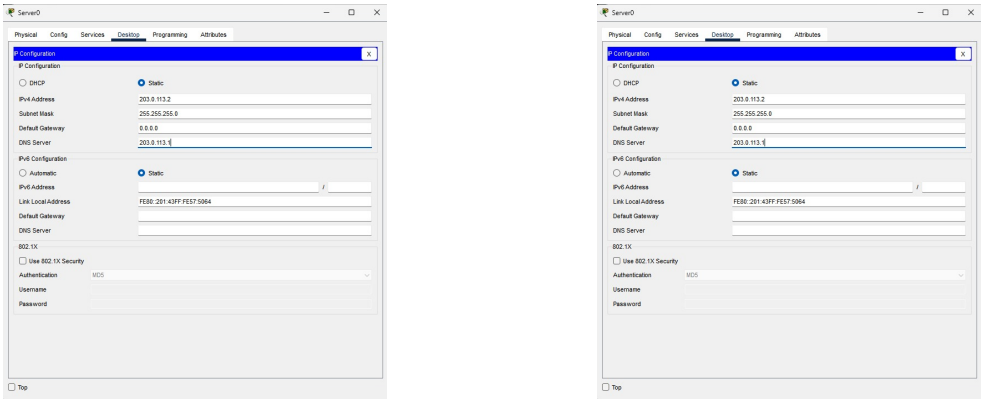


Gambar 11: Topologi awal jaringan sebelum konfigurasi.

2. **Konfigurasi Alamat IP Perangkat:** Setiap perangkat dikonfigurasi dengan alamat IP statis sesuai skema pengalamatan yang telah dirancang. Hal ini penting agar setiap perangkat memiliki identitas yang jelas di dalam jaringannya masing-masing dan untuk mempermudah proses pembuatan aturan firewall nanti.

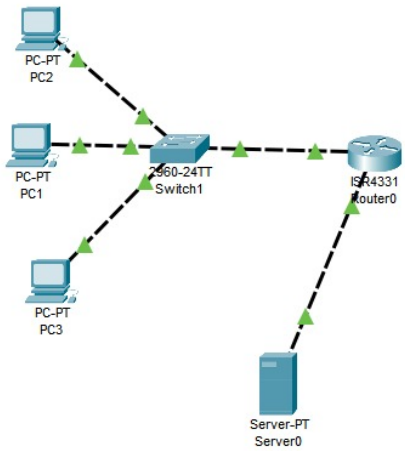


(a) IP PC1 (b) IP PC2 (c) IP PC3



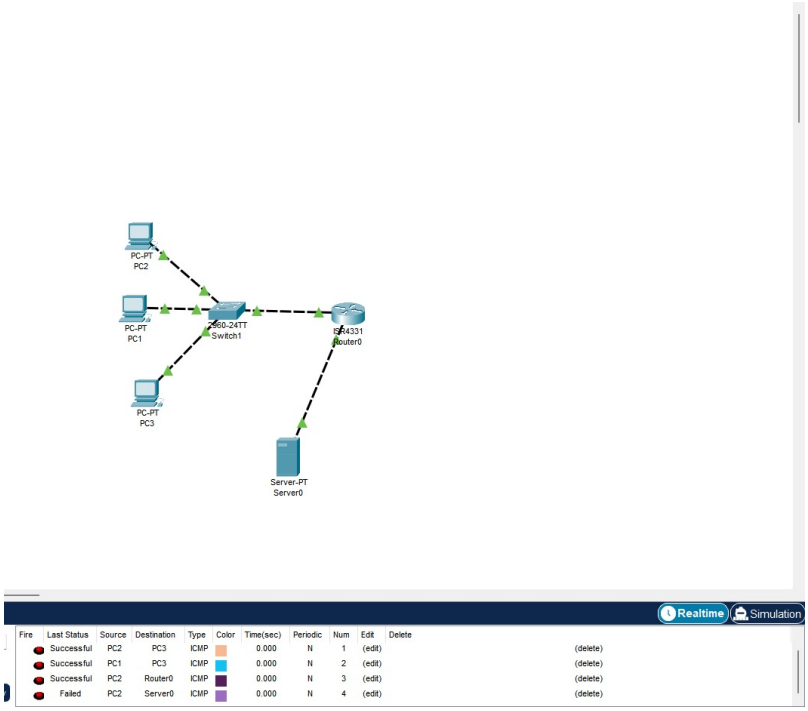
(d) IP Server (e) IP Router via CLI

Gambar 12: Kolase konfigurasi IP pada (a) PC1, (b) PC2, (c) PC3, (d) Server, dan (e) antarmuka Router.



3. **Verifikasi Konektivitas Awal:** Setelah semua IP diatur, saya melakukan uji `ping` untuk memverifikasi konektivitas. Sesuai asumsi, ping antar PC dalam satu LAN berhasil karena mereka terhubung melalui switch pada segmen jaringan yang sama. Namun, ping dari PC ke Server gagal. Ini terjadi karena belum ada NAT; server di

jaringan publik tidak tahu cara mengirim balasan ke alamat IP privat `192.168.10.2`.

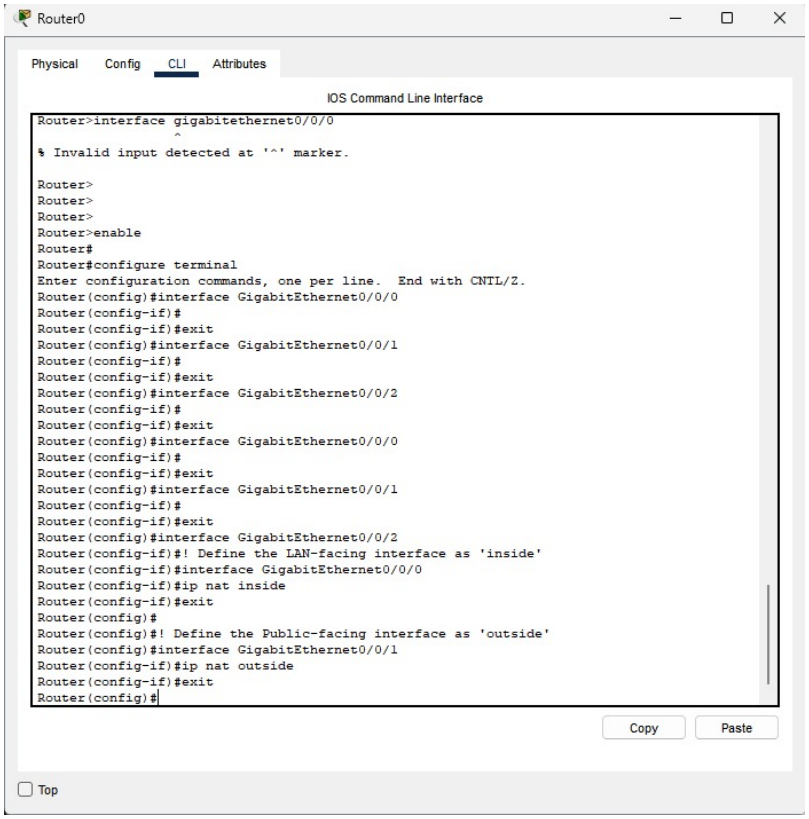


Gambar 13: Hasil ping: sukses antar PC di LAN, gagal dari PC ke Server.

- **Implementasi NAT Overload**

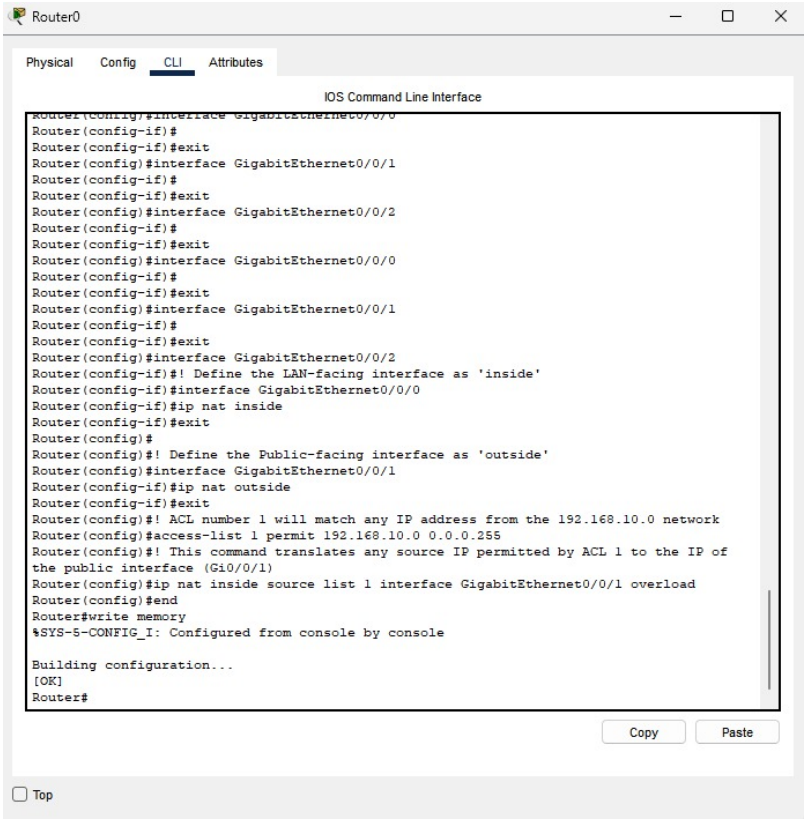
Untuk mengatasi masalah konektivitas ke server, saya mengimplementasikan NAT Overload (PAT).

5. **Definisi Antarmuka NAT:** Langkah pertama adalah memberitahu router mana antarmuka yang menghadap ke dalam (LAN) dan mana yang ke luar (Publik). Saya menetapkan `GigabitEthernet0/0/0` sebagai `ip nat inside` dan `GigabitEthernet0/0/1` sebagai `ip nat outside`.



Gambar 14: Konfigurasi antarmuka NAT inside dan outside.

6. **Pembuatan Aturan NAT:** Saya membuat ACL (`access-list 1`) untuk mengidentifikasi trafik dari LAN yang boleh ditranslasikan, kemudian mengikatnya ke antarmuka publik dengan perintah `ip nat inside source list 1 interface ... overload`. Ini adalah inti dari konfigurasi NAT.



Gambar 15: Perintah untuk membuat ACL dan aturan NAT Overload.

7. **Pengujian Setelah NAT:** Setelah NAT aktif, saya kembali melakukan uji `ping` dari setiap PC ke Server. Hasilnya, semua ping berhasil. Ini membuktikan bahwa NAT telah bekerja dengan benar, menerjemahkan alamat IP privat setiap PC menjadi alamat IP publik router.

• Skenario Firewall 1: Izinkan Hanya PC1

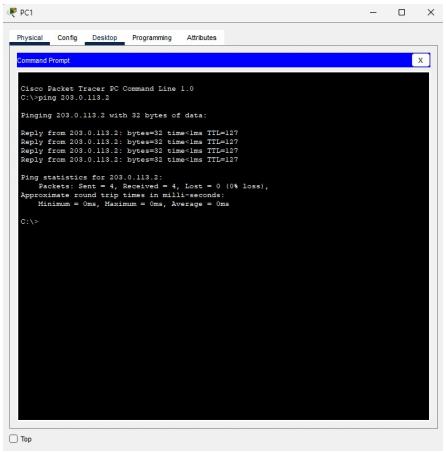
Skenario pertama adalah menerapkan kebijakan keamanan di mana hanya PC1 yang diizinkan mengakses Server.

8. **Pembuatan ACL Skenario 1:** Saya membuat `access-list 20` yang hanya berisi satu baris perintah: `permit host 192.168.10.2`. ACL ini memiliki sifat "implicit deny" di akhir, yang berarti semua trafik lain yang tidak cocok dengan aturan permit ini akan otomatis ditolak. Aturan ini kemudian saya terapkan secara *inbound* pada antarmuka LAN router.

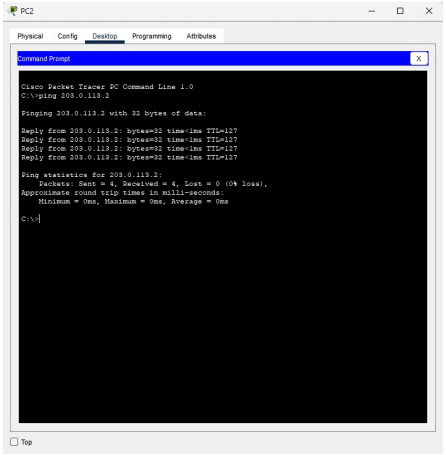
```
Router(config-if)#access-list 20 permit host 192.168.10.2
Router(config)#interface GigabitEthernet0/0/0
Router(config-if)#ip access-group 20 in
Router(config-if)#exit
Router(config)#end
Router#write memory
%SYS-5-CONFIG_I: Configured from console by console

Building configuration...
[OK]
Router#
```

Gambar 17: Konfigurasi ACL 20 untuk mengizinkan hanya PC1.

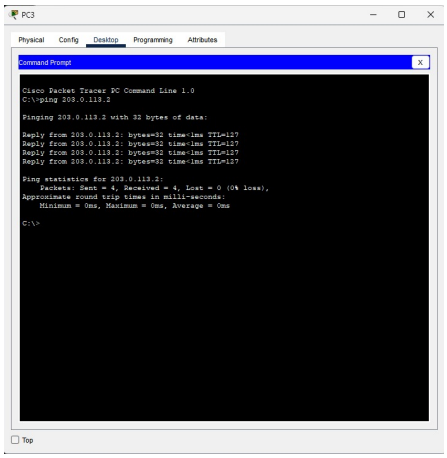


(a) Ping PC1 ke Server



(b) Ping PC2 ke Server

please speed i need this



(c) Ping PC3 ke Server

Gambar 16: Uji konektivitas setelah NAT aktif menunjukkan semua PC dapat terhubung ke Server.

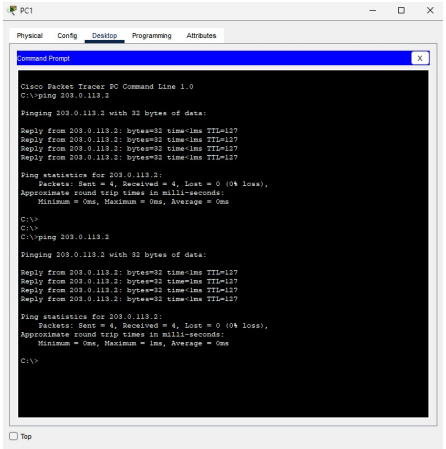
9. **Pengujian Skenario 1:** Hasil pengujian sesuai dengan yang diharapkan. Ping dari PC1 berhasil, sementara ping dari PC2 dan PC3 gagal. Uji ping antar PC di LAN juga tetap berhasil, membuktikan bahwa ACL yang diterapkan pada router tidak mengganggu komunikasi lokal yang tidak melewati router.

• Skenario Firewall 2: Blokir PC2 dan PC3

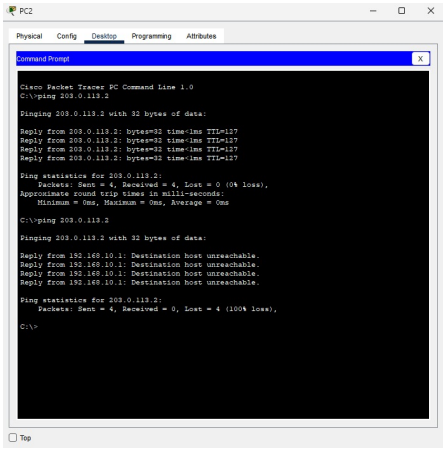
Skenario kedua adalah menerapkan kebijakan yang secara eksplisit memblokir PC2 dan PC3.

10. **Perubahan Konfigurasi ACL:** Pertama, saya membersihkan konfigurasi ACL sebelumnya. Kemudian saya membuat `access-list 30` dengan urutan: dua baris `deny` untuk PC2 dan PC3, diikuti satu baris `permit any` untuk mengizinkan semua trafik lainnya (termasuk PC1). Proses ini diverifikasi menggunakan perintah `show`.

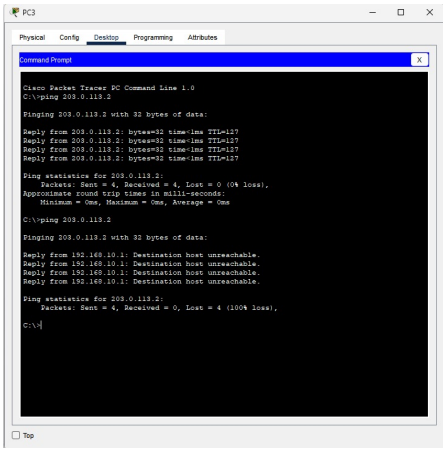
11. **Pengujian Skenario 2:** Pengujian akhir dilakukan dan hasilnya kembali sesuai de-



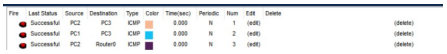
(a) PC1 ke Server (Success)



(b) PC2 ke Server (Failed)



(c) PC3 ke Server (Failed)



(d) PC2 ke PC3 (Success)

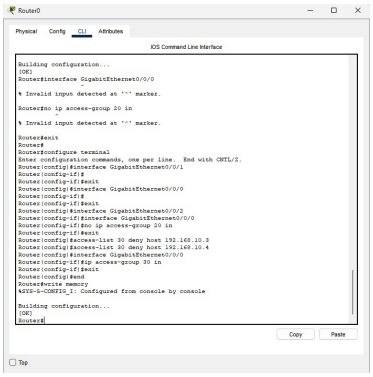
Gambar 18: Hasil pengujian ACL Skenario 1.

ngan konfigurasi. PC1 berhasil terhubung karena trafiknya lolos dari aturan `deny` dan diizinkan oleh `permit any`. Sementara itu, PC2 dan PC3 gagal karena trafik mereka cocok dengan aturan `deny` pertama yang ditemui oleh router.

4 Kesimpulan

Melalui serangkaian percobaan dan simulasi pada praktikum ini, saya telah berhasil mengimplementasikan dan menganalisis dua konsep inti jaringan komputer, yaitu Network Address Translation (NAT) dan Firewall. Saya memahami bahwa NAT, khususnya melalui metode PAT (‘masquerade‘ pada MikroTik dan ‘overload‘ pada Cisco), merupakan solusi fundamental untuk mengatasi kelangkaan alamat IPv4 dan berfungsi sebagai gerbang utama bagi jaringan lokal untuk dapat mengakses internet secara bersamaan. Implementasi yang berhasil pada kedua platform menunjukkan peran universal dan krusial dari teknologi ini dalam arsitektur jaringan modern.

Selanjutnya, pada topik Firewall, saya mendapatkan pemahaman mendalam menge-



(a) Hapus ACL lama

```
Router(config)#interface GigabitEthernet0/0/0
Router(config-if)#ip nat inside
Router(config-if)#exeit
                  ^
% Invalid input detected at '^' marker.

Router(config-if)#exit
Router(config)#interface GigabitEthernet0/0/1
Router(config-if)#ip nat outside
Router(config-if)#exit
Router(config)#
```

(b) Buat ACL baru

```
Router(config-if)#exit
Router(config)#interface GigabitEthernet0/0/1
Router(config-if)#ip nat outside
Router(config-if)#exit
Router(config)#access-list 1 permit 192.168.10.0 0.0.0.255
Router(config)#ip nat inside source list 1 interface GigabitEthernet0/0/1 overload
Router(config)#access-list 30 deny host 192.168.10.3
Router(config)#access-list 30 deny host 192.168.10.4
Router(config)#access-list 30 permit any
Router(config)#interface GigabitEthernet0/0/0
Router(config-if)#ip access-group 30 in
Router(config-if)#exit
Router(config)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#
```

(c) Terapkan ACL

```
Router#show access-lists
Standard IP access list 20
    10 permit host 192.168.10.2 (5 match(es))
Standard IP access list 1
    10 permit 192.168.10.0 0.0.0.255
Standard IP access list 30
    10 deny host 192.168.10.3
    20 deny host 192.168.10.4
    30 permit any

Router#
```

(d) Verifikasi ‘show access-lists‘

```
Router#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0/0   192.168.10.1    YES manual up                    up
GigabitEthernet0/0/1   203.0.113.1     YES manual up                    up
GigabitEthernet0/0/2   unassigned      YES unset  administratively down down
Vlan1                  unassigned      YES unset  administratively down down
Router#
```

(e) Verifikasi ‘show ip int brief‘

Gambar 19: Langkah-langkah perubahan konfigurasi ACL untuk Skenario 2.

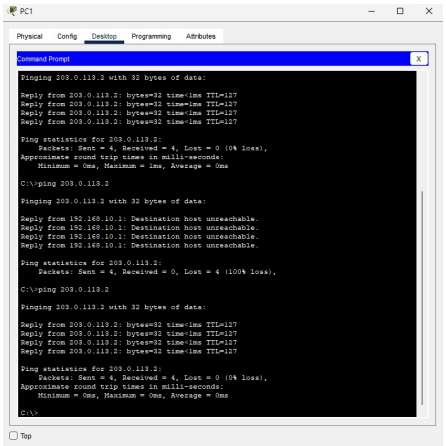
nai pentingnya keamanan berlapis. Melalui praktikum MikroTik, saya belajar cara memfilter lalu lintas tidak hanya berdasarkan protokol Layer 4 seperti ICMP, tetapi juga hingga ke Layer 7 dengan memblokir akses berdasarkan konten spesifik dari sebuah situs web. Sementara itu, pada Tugas Modul simulasi Cisco, saya mendalami penggunaan Access Control List (ACL) sebagai fondasi firewall. Dari simulasi tersebut, saya dapat menyimpulkan bahwa efektivitas sebuah ACL sangat bergantung pada logika dan urutan aturan (‘rule order‘), pemahaman akan ‘implicit deny‘ yang secara default menolak semua lalu lintas yang tidak diizinkan secara eksplisit, serta penempatan aturan (‘inbound‘ atau ‘outbound‘) yang strategis untuk mengontrol alur data secara efisien tanpa mengganggu komunikasi internal di dalam LAN.

Secara keseluruhan, praktikum ini memberikan pengetahuan praktis yang komprehensif, membuktikan bahwa jika NAT adalah kunci untuk konektivitas, maka Firewall yang terkonfigurasi dengan baik adalah pilar keamanannya, di mana keduanya harus bekerja secara sinergis untuk menciptakan lingkungan jaringan yang fungsional sekaligus aman.

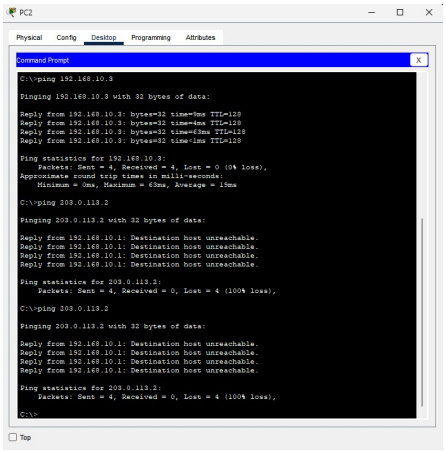
5 Lampiran

Dokumentasi saat praktikum

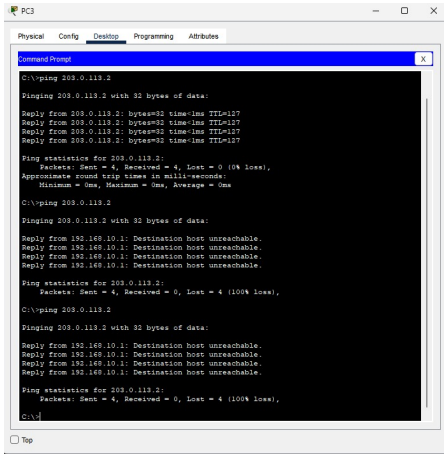
Berikut adalah dokumentasi saat praktikum Modul 4 berlangsung



(a) PC1 ke Server (Success)

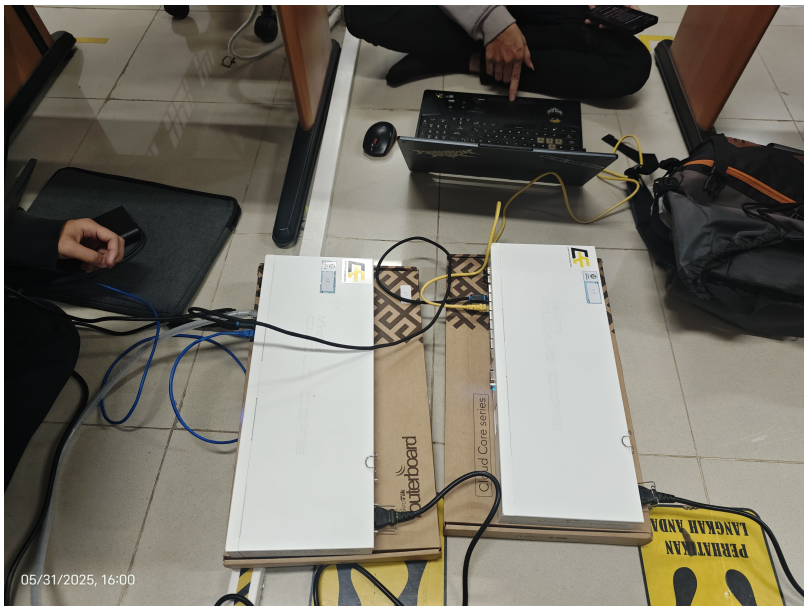


(b) PC2 ke Server (Failed)



(c) PC3 ke Server (Failed)

Gambar 20: Hasil pengujian akhir setelah ACL diubah untuk memblokir PC2 dan PC3.



Gambar 21: Dokumentasi.