

# Modul Prapraktikum

## IF2230 Jaringan Komputer

4 - DNS, SSH, Network Access Control List, dan IPv6

Dipersiapkan oleh:

妹ラボラトリー

*(Asisten Laboratorium Sistem Terdistribusi)*

Sister; Lab<sup>22</sup>

**START**

Senin, 28 April 2025, 16.00 WIB

**END**

Senin, 5 Mei 2025, 12.00 WIB

*Estimasi waktu pengerjaan: ± 7 jam*

# Daftar Revisi

## Selasa, 29 April, 2025

1. Perbaikan penulisan tugas di [Tugas 1](#)
2. Perbaikan penulisan tugas di [Tugas 3](#)
3. Perbaikan deskripsi di [Tugas 7](#).

## Rabu, 30 April, 2025

1. Perbaikan tanggal *deadline*.

## Sabtu, 3 Mei, 2025

1. Penambahan tautan [cheat sheet](#) (akan diperbolehkan untuk dibuka saat praktikum 4).
2. Penambahan [referensi](#).

# Latar Belakang

Tugas ini ditujukan untuk mempersiapkan peserta untuk praktikum terakhir kuliah ini. Dengan menyelesaikan tugas ini, praktikan diharapkan memiliki persiapan dan pengetahuan dasar terhadap materi yang dibutuhkan.

Berikut topik-topik yang menjadi lingkup modul ini:

- DNS
- SSH
- Network Access Control Lists
- IPv6

# Peraturan

Kerjakan tugas ini dengan mengikuti peraturan-peraturan yang sama dengan [prapraktikum sebelumnya](#).

# Pengerjaan dan Deliverables

Kerjakan dan kumpulkan tugas ini dengan mengikuti semua ketentuan berikut.

1. Simpan tugas Anda dengan format ini: **IF2230\_PraPrak[X]\_<NIM>.pdf**  
(contoh: IF2230\_PraPrak4\_13522022.pdf)
2. Kumpulan tugas Anda melalui [form ini](#).
3. Tenggat waktu untuk tugas prapraktikum ini adalah **Senin, 5 Mei 2025, pukul 12.00 WIB**.
4. Ketentuan pengerjaan dan *deliverables* lainnya sama dengan [prapraktikum sebelumnya](#).
5. Q&A: [Link QnA](#).

## Modul Prapraktikum

### DNS

Dalam praktikum sebelumnya, kita sudah mengeksplorasi topik alamat IP, *routing*, dan translasi alamat. Namun, manusia mengakses informasi secara online melalui nama yang lebih mudah dibaca seperti `google.com`. Sistem ini disebut **domain name system (DNS)**. Server DNS menghilangkan kebutuhan bagi manusia untuk mengingat alamat IP seperti `192.168.1.1`.

Proses resolusi DNS melibatkan **konversi nama host menjadi alamat IP yang dapat dibaca oleh komputer**. Sebuah translasi harus terjadi antara permintaan pengguna hingga alamat yang dapat dibaca oleh mesin yang diperlukan untuk menemukan sumber daya yang ditetapkan pada nama host oleh pemilik domain melalui pendaftar domain ([domain registrar](#)) diterima.

Ada 8 langkah dalam pencarian DNS:

1. Seorang pengguna mengetik `example.com` ke dalam aplikasi klien dan *query* tersebut menjelajahi internet hingga akhirnya diterima oleh **DNS recursive resolver**.

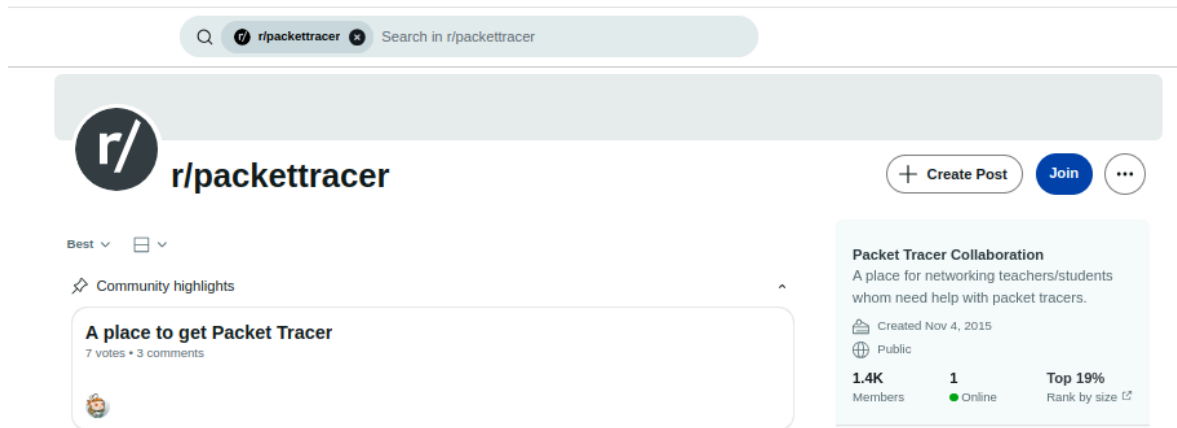
2. Resolver kemudian menanyakan DNS root name server (.).
3. DNS root name server kemudian merespons resolver dengan alamat server DNS Top Level Domain (TLD) (contoh .com atau .net), yang menyimpan informasi untuk domain-domainnya. Ketika mencari example.com, permintaan kita diarahkan menuju TLD .com.
4. Resolver kemudian mengajukan permintaan ke TLD .com.
5. Server TLD kemudian merespons dengan alamat IP dari nameserver domain, example.com.
6. Terakhir, resolver rekursif mengirim query ke nameserver milik domain example.com.
7. Alamat IP untuk example.com kemudian dikembalikan ke resolver dari nameserver.
8. Resolver DNS kemudian merespons klien dengan alamat IP dari domain yang diminta sebelumnya.

**Entri DNS mungkin disimpan dalam router lokal, browser, atau sistem operasi.** Ini membuat akses lebih cepat dengan melewati langkah-langkah pencarian, tetapi juga memungkinkan serangan seperti peracunan DNS ([DNS poisoning](#)).

Tugas 0	
Q	<p>Pada tugas ini, kita akan mengeksplorasi <code>nslookup</code>, salah satu program untuk melakukan pencarian domain name yang seharusnya sudah ada <i>by default</i> pada semua komputer.</p> <p>Bukalah sebuah terminal (i.e. <i>command prompt</i>, <i>bash</i>, etc) di komputer pribadi Anda. Jalankan <code>nslookup itb.ac.id</code> untuk menentukan IPv4 Address <code>itb.ac.id</code>.</p> <p>Tugas: tuliskan alamat IP <code>itb.ac.id</code></p>
A	34.36.102.170
Q	<p>Gunakan <code>nslookup</code> untuk menentukan <i>nameserver</i> <code>sisterlab.tech</code>.</p> <p>Hint: gunakan opsi <code>-type=soa</code></p> <p>Tugas: tuliskan <i>nameserver</i> <code>sisterlab.tech</code></p>

A	nsid1.rumahweb.com
Q	<p>Gunakan <code>nslookup</code> untuk menentukan domain name <code>146.112.62.105</code>.</p> <p>Tugas: tuliskan domain name <code>146.112.62.105</code></p>
A	opendns.com
Q	<p>Jalankan <i>command</i> <code>nslookup -debug reddit.com 1.1.1.1</code>. Apa yang terjadi?</p> <p><b>Note:</b> tugas ini berasumsi reddit diblokir oleh ISP yang Anda gunakan. Jika reddit tidak diblokir, tentukan dan gantilah reddit dengan situs lain yang diblokir.</p> <p>Tugas: jelaskan hasil <i>command</i>. Penjelasan Anda boleh panjang maupun pendek, tetapi setidaknya harus menjawab:</p> <ol style="list-style-type: none"> <li>1. Apakah hasil menunjukkan alamat IP reddit yang sesungguhnya?</li> <li>2. Apa alamat IP apa yang dikembalikan (dan domain name nya)?</li> <li>3. Mengapa itu terjadi?</li> </ol> <p>Kemudian ketika Anda menggunakan <i>browser</i>, DNS <i>provider</i> yang digunakan oleh <i>browser</i> bergantung pada pengaturannya. Browser yang Anda gunakan kemungkinan besar menyediakan opsi untuk menggunakan <i>default</i> DNS <i>provider</i> (yaitu DNS <i>provider</i> yang sama dengan yang digunakan pada <code>nslookup</code>), serta opsi untuk menggunakan DoH (<a href="#">DNS over HTTPS</a>).</p> <p>Tugas: ubah pengaturan salah satu <i>browser</i> yang Anda gunakan untuk bisa mengakses <code>https://www.reddit.com/r/packettracer/</code> tanpa diblokir oleh ISP. Berikan tangkapan layar halaman utamanya. Kemudian, jelaskan bagaimana cara kerja DNS over HTTPS dan bagaimana penggunaannya bisa mem-<i>bypass content filter</i> yang sudah di-set oleh ISP.</p> <p>Untuk referensi, Anda bisa membaca halaman pada <a href="#">tautan ini</a>.</p>
A	<ol style="list-style-type: none"> <li>1. Bukan, alamat yang dikembalikan adalah IP Address server yang menampilkan website pemblokiran/internet positif milik Telkomsel</li> <li>2. 202.3.218.137 dengan domain name mypage.blocked.bltsel</li> <li>3. Hal ini terjadi ketika saya menggunakan operator telkomsel dan memilih DNS Provider secara otomatis (umumnya menggunakan DNS Server milik operator) sehingga ketika saya melakukan browsing reddit.com akan melakukan DNS Request kepada DNS Server Telkomsel dan kebijakan dari Telkomsel untuk memblokir website reddit membuat respon dari DNS Server bukan</li> </ol>

merupakan IP Address milik reddit melainkan IP Address server web blocking/internet positif sehingga seolah-olah pengguna diarahkan ke web tersebut dan tidak bisa mengakses reddit.com



DNS over HTTPS mencegah admin jaringan maupun ISP untuk melihat dan melakukan perubahan saat pengguna melakukan DNS request. DoH bekerja dengan melakukan resolve DNS menggunakan protokol HTTPS melalui port 443 menuju DNS server yang diinginkan. Dengan begini pengguna bisa melakukan bypass content filtering yang diterapkan oleh penyedia jaringan.

Menambahkan *resource* di server DNS dilakukan dengan menambahkan entri DNS yang mungkin bervariasi tergantung pada platform DNS yang digunakan. Cloudflare adalah salah satu penyedia layanan pendaftaran DNS yang paling dikenal dengan alamat server DNS 1.1.1.1. Tahukah Anda? Menurut [Forbes](#), **Matthew Prince**, CEO Cloudflare, memiliki *net worth* sebesar USD 2,3 miliar per Maret 2023, menjadikannya orang terkaya kedua di Utah setelah Gail Miller. Gambar di bawah ini menunjukkan antarmuka manajemen catatan DNS Cloudflare.

DNS management for [REDACTED]

Review, add, and edit DNS records. Edits will go into effect once saved.

DNS Setup: Full ⓘ Import and Export ▼ ⚙ Dashboard Display Settings

Search DNS Records

[Add filter](#)  [Search](#) [Add record](#)

[name] points to [IPv4 address] and has its traffic proxied through Cloudflare.

Type: A Name (required): [REDACTED] IPv4 address (required): [REDACTED] Proxy status: ☒ Proxied TTL: Auto

Record Attributes [Documentation](#)

The information provided here will not impact DNS record resolution and is only meant for your reference.

Comment

[Cancel](#) [Save](#)

Type ⓘ	Name ⓘ	Content ⓘ	Proxy status ⓘ	TTL ⓘ	Actions
<input type="checkbox"/> A	[REDACTED]	[REDACTED]	Proxied	Auto	<a href="#">Edit</a>
<input type="checkbox"/> A	[REDACTED]	[REDACTED]	DNS only	Auto	<a href="#">Edit</a>
<input type="checkbox"/> A	[REDACTED]	[REDACTED]	DNS only - reserved IP	Auto	<a href="#">Edit</a>
<input type="checkbox"/> A	[REDACTED]	[REDACTED]	DNS only	Auto	<a href="#">Edit</a>
<input type="checkbox"/> A	[REDACTED]	[REDACTED]	Proxied	Auto	<a href="#">Edit</a>
<input type="checkbox"/> AAAA	[REDACTED]	[REDACTED]	DNS only	Auto	<a href="#">Edit</a>
<input type="checkbox"/> CNAME	[REDACTED]	[REDACTED]	Proxied	Auto	<a href="#">Edit</a>
<input type="checkbox"/> MX	[REDACTED]	[REDACTED]	DNS only	Auto	<a href="#">Edit</a>
<input type="checkbox"/> TXT	[REDACTED]	[REDACTED]	DNS only	Auto	<a href="#">Edit</a>
<input type="checkbox"/> TXT	[REDACTED]	[REDACTED]	DNS only	Auto	<a href="#">Edit</a>

Gambar 1. Cloudflare DNS record management interface

Begitu pula, menambahkan entri catatan DNS di server DNS Cisco Packet Tracer dapat dilakukan di antarmukanya.

Server1

Physical Config **Services** Desktop Programming Attributes

**SERVICES**

- HTTP
- DHCP
- DHCPv6
- TFTP
- DNS**
- SYSLOG
- AAA
- NTP
- EMAIL
- FTP
- IoT
- VM Management
- Radius EAP

DNS

DNS Service ☐ On ☒ Off

Resource Records

Name  Type A Record

Address

[Add](#) [Save](#) [Remove](#)

No.	Name	Type	Detail
-----	------	------	--------

Gambar 2. Antarmuka manajemen catatan DNS server Cisco Packet Tracer

Ada berbagai jenis catatan DNS ([DNS records](#)), di antaranya adalah catatan A, catatan AAAA, dan CNAME. Catatan A menyimpan alamat IP dari sebuah domain; catatan AAAA

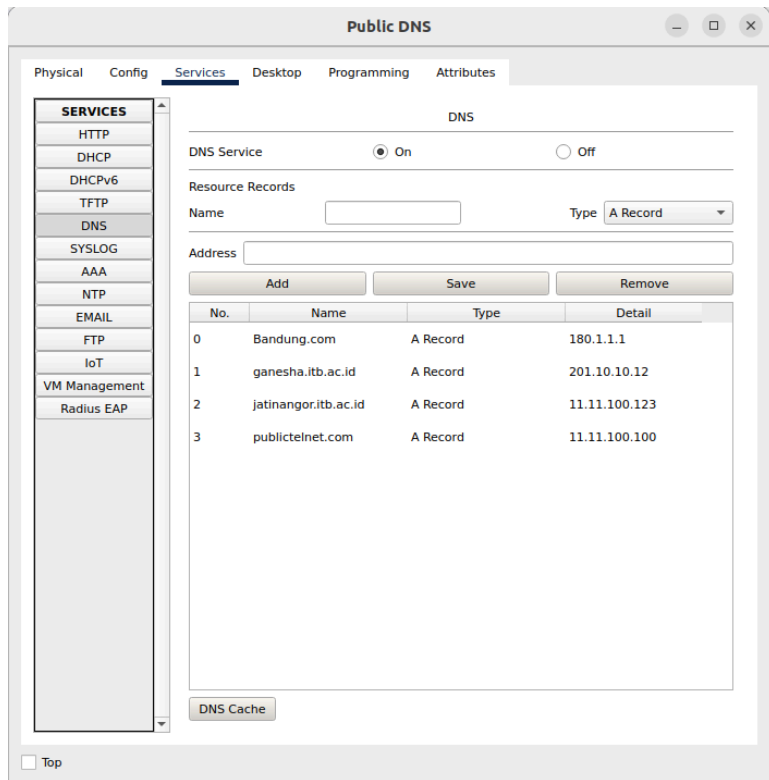
menyimpan alamat IPv6 untuk sebuah domain, berbeda dengan catatan A yang mencantumkan alamat IPv4; sedangkan catatan CNAME meneruskan satu domain atau subdomain ke domain lain, bukan memberikan alamat IP. Soal teori tentang catatan domain  **mungkin** akan ditanyakan sebagai salah satu soal teori pada praktikum, jadi silakan baca referensi yang telah diberikan di atas.

*Apakah Anda pernah mempertimbangkan untuk membeli domain sendiri untuk mengarah ke server Anda sendiri? Beberapa [domain](#) sangat murah, dengan yang termurah di Indonesia harganya tidak sampai dua dolar per tahun!*

Tugas 1	
Q	<p>Unduhlah kembali <a href="#">NAT-start-v2.pkt</a> ↓.</p> <p>Di server DNS dengan alamat yang terdaftar dalam konfigurasi Anda, tambahkan sebuah DNS <i>record</i> dengan <code>ganesha.itb.ac.id</code> sebagai nama domain, yang menunjuk ke alamat Ganesha Server (alamat publik).</p> <p>Tugas: Tampilkan tabel DNS yang baru. Setelah mengatur DNS, coba akses <code>ganesha.itb.ac.id</code> dari Public PC (tentu saja melalui <i>browser</i>)! Terakhir, amati paket yang dikirim untuk permintaan web tersebut, dan jelaskan prosesnya!</p>



A



1. DNS Client Public PC mengirimkan DNS Request via UDP ke Public DNS Server
2. DNS Server mencari IP Address yang berasosiasi oleh domain ganesha.itb.ac.id dan mengirimkan respons hasil ke Public PC via UDP
3. Public PC menerima respons dari DNS Server berisi IP Address Ganesha Server dan mencoba membangun koneksi TCP kepada Ganesha Server (SYN)
4. Ganesha Server merespons dengan mengirimkan SYN + ACK segment kepada Public PC
5. Public PC menerima respons Ganesha Server dan melakukan HTTP Request ke Ganesha Server
6. Ganesha Server menerima HTTP Request dan mengembalikan HTTP Reply ke Public PC
7. Selanjutnya koneksi TCP akan diterminasi setelah semua paket selesai dikirim dan diterima

## SSH

Secure Shell ([SSH](#)) adalah metode untuk mengirimkan perintah secara aman ke komputer melalui jaringan yang tidak aman, biasanya untuk mengontrol server dari jarak jauh, mengelola infrastruktur, dan untuk mentransfer file. SSH menggunakan [kriptografi](#)

untuk mengautentikasi dan mengenkripsi koneksi antara perangkat. Port default untuk SSH adalah 22.

SSH aman karena mengintegrasikan enkripsi dan autentikasi melalui proses yang disebut [public key cryptography](#). *Public key cryptography* adalah cara untuk mengenkripsi data menggunakan *asymmetric key*. Salah satu key, yaitu *public key*, tersedia untuk digunakan siapa saja. Key lainnya, yaitu *private key*, disimpan rahasia oleh pemiliknya. Karena kedua key saling berhubungan, penetapan identitas pemilik key memerlukan kepemilikan *private key* yang sesuai dengan *public key*, karena **data yang dienkripsi dengan public key memerlukan private key untuk dekripsi dan sebaliknya**. Dalam koneksi SSH, kedua pihak memiliki sepasang public/private key, dan masing-masing pihak mengautentikasi satu sama lain menggunakan key-key ini.

*Asymmetric key* ini memungkinkan kedua pihak dalam koneksi untuk melakukan *randomization* sebuah *symmetric key* yang identik dan bersama untuk enkripsi lebih lanjut melalui saluran tersebut. **Setelah negosiasi ini selesai, kedua pihak menggunakan symmetric key untuk mengenkripsi data yang mereka tukar karena enkripsi dan dekripsi menggunakan symmetric key lebih cepat.** Terdapat banyak algoritma enkripsi dan lebih banyak rinciannya mengenai kriptografi yang akan dibahas dalam IF4020 Kriptografi. Salah satu algoritma yang paling menonjol digunakan untuk enkripsi asimetris adalah [RSA](#) dengan panjang key minimum 2048. Sementara itu, [AES](#) dengan panjang key 256 digunakan untuk enkripsi simetris.

Untuk mengonfigurasi server SSH di router Cisco, diperlukan nama host yang bukan default dan nama domain. Atur nama host dan nama domain menggunakan perintah di bawah ini.

```
Router(config)# hostname <name>
Router(config)# ip domain-name <name>
```

Setelah itu, dibutuhkan pengguna dan kata sandi. Perintah berikut dapat digunakan untuk membuat pengguna di router.

```
Router(config)# username <name> privilege <level> secret <password>
```

Tingkat hak akses menentukan perintah yang akan diotorisasi untuk dijalankan oleh pengguna pada perangkat; penjelasan lebih lanjut dapat diakses di sini. Ketahui bahwa 1 adalah yang terendah dan 15 adalah yang tertinggi. Untuk saat ini, gunakan 15 untuk akses SSH. Enkripsi dan versi SSH juga dapat dikonfigurasi, tetapi PC harus mendukung algoritma

hash dan versi SSH yang dipilih, yang seringkali menambah tantangan dalam penerapannya. Untuk menyederhanakan, biarkan semuanya pada default.

Setelah membuat pengguna, generate key yang akan digunakan untuk pertukaran key. Perintah yang digunakan untuk membuat key adalah sebagai berikut, dengan prompt yang meminta spesifikasi lebih lanjut pada implementasi yang mungkin diperlukan tergantung pada pilihan algoritma Anda.

```
Router(config)# crypto key generate <algorithm>
```

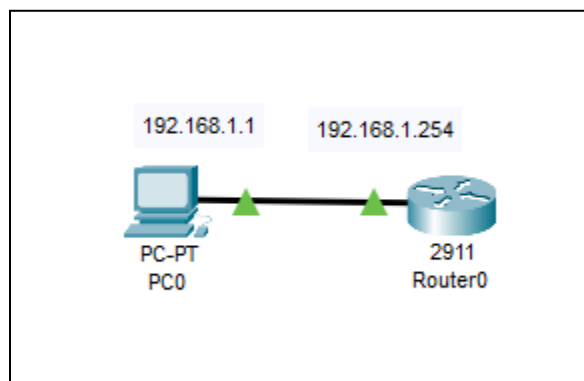
Terakhir, atur sebuah line untuk memungkinkan SSH ke dalam router.

```
Router(config)# line vty 0 15
Router(config-line)# login local
Router(config-line)# transport input ssh
```

Dengan cara ini, koneksi SSH dapat dipertahankan menggunakan key yang disimpan di router.

## Tugas 2

Q Hubungkan sebuah PC dan *router* pada Cisco Packet Tracer seperti berikut.



- Siapkan server SSH di router dengan username 'cisco' dan password 'cisco'.
- Kemudian, sambungkan ke router menggunakan SSH dari PC0.
- Jalankan perintah `show ip interface brief` dari koneksi SSH.
- Jalankan perintah `exit` untuk menutup koneksi SSH.

Tugas: cantumkan hasil dari perintah `show ip interface brief` dan `exit` dari koneksi SSH di area di bawah ini.

A

```
R1#show ip interface brief
Interface      IP-Address      OK? Method Status      Protocol
GigabitEthernet0/0  192.168.1.254  YES manual  up          up
GigabitEthernet0/1  unassigned      YES unset   administratively down down
GigabitEthernet0/2  unassigned      YES unset   administratively down down
Vlan1          unassigned      YES unset   administratively down down
R1#exit

[Connection to 192.168.1.254 closed by foreign host]
```

## Tambahan Terkait SSH (Tidak Diujikan, Tetapi Penting)

**Materi SSH di bagian ini berada di luar cakupan Cisco Packet Tracer.** Sebenarnya, cara yang lebih tepat untuk mengaktifkan SSH ke router dari PC adalah dengan menghasilkan sepasang key di komputer dan kemudian memasukkan public key yang dihasilkan ke router. Ini dapat dilakukan dengan memasuki mode konfigurasi keychain, menentukan pengguna yang terkait dengan key, dan memasukkan key-string.

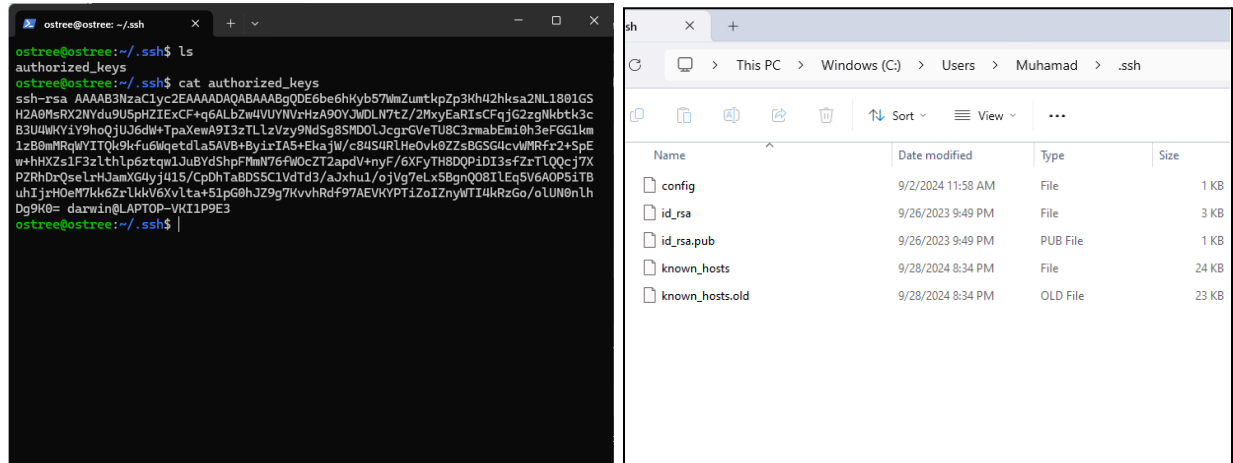
```
Router(config)# ip ssh pubkey-chain
Router(conf-ssh-pubkey)# username <name>
Router(conf-ssh-pubkey-user)# key string
Router(conf-ssh-pubkey-data)# <key, 254 chars at a time>
Router(conf-ssh-pubkey-data)# exit
```

Langkah ini memungkinkan login tanpa kata sandi jika mode autentikasi SSH mengizinkan penggunaan *public key*. Login tanpa kata sandi lebih aman dari serangan *social engineering*, dan penyimpanan key di perangkat pengguna memastikan hanya pengguna yang sah yang boleh mengakses. Untuk memperkuat lagi, **autentikasi berbasis login menggunakan password bisa dinonaktifkan** sehingga pengguna tanpa *public key* tidak bisa login.

```
Router(config)# no ip ssh server authenticate user password
```

**Sayangnya, Cisco Packet Tracer memiliki batasan. Tidak mungkin untuk menghasilkan sepasang key di PC Cisco Packet Tracer maupun untuk mengonfigurasi keychain router.** Namun, ini adalah pengetahuan yang baik karena koneksi ke server dilakukan dengan cara yang sama dengan menyalin *public key* Anda ke dalam keychain pengguna. Di openssh GNU/Linux, itu berada di folder `~/.ssh/` dengan key akses publik terletak di file `authorized_keys`. Sementara di Windows, itu terletak di `%USER%.ssh`.

Jika Anda tertarik: [Tutorial on how to do SSH Public Keys on actual PCs](#)🔗



Gambar 3. openssh keychain (kiri), Windows SSH keychain (kanan)

Ini mungkin tidak digunakan dalam praktikum kita, tetapi *in practice*, kemungkinan besar Anda akan sering menggunakan ini.

Quick Exercise: is the system above being endangered due to exposing the public key?  
The answer is obviously **no**, but can you explain why?

# Network Access Control List

**Network access-control list (ACL)** adalah sekumpulan aturan yang mengizinkan atau menolak akses ke sumber daya komputer pada tingkat jaringan. *Network access control list* dapat berfungsi sebagai penyaring paket yang menginstruksikan *router* untuk mengizinkan atau membuang *traffic* tertentu. ACL dapat menyaring lalu lintas berdasarkan alamat IP sumber/tujuan, port sumber/tujuan, protokol, dan lain sebagainya. **ACL dievaluasi dari atas ke bawah** sehingga jika suatu permintaan memenuhi beberapa aturan dalam ACL, aturan teratas akan diterapkan. Di sisi lain, jika suatu permintaan tidak memenuhi aturan mana pun dalam ACL, maka akan jatuh pada aturan default yang diterapkan pada ACL. Aturan default dalam ACL bisa berupa penolakan implisit di mana paket selalu dibuang atau pengizinan implisit di mana paket selalu diizinkan.

Secara umum, ada dua jenis ACL IP di router Cisco, yaitu **ACL IP standar** dan **ACL IP extended**. ACL IP standar hanya **menyaring traffic berdasarkan alamat IP sumber paket**, sedangkan ACL IP *extended* dapat **menyaring traffic berdasarkan lebih banyak parameter seperti protokol, port, dan alamat sumber serta tujuan**.

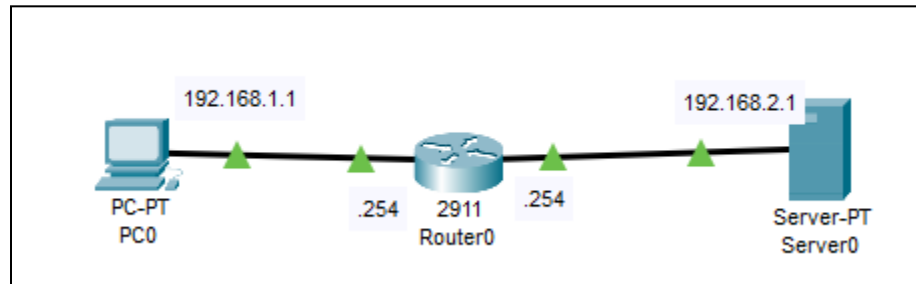
Mengonfigurasi ACL IP standar dan diperluas di router Cisco dapat dilakukan dengan secara berurutan membuat entri daftar akses bernomor atau bernama dan kemudian menetapkan pada suatu *interface*. Misalnya, entri ACL IP standar yang sederhana dapat ditambahkan di mode konfigurasi global dengan menjalankan perintah berikut.

```
RouterA(config)# access-list <number> {permit | deny} <IP> <Wildcard Mask>
```

Dengan parameter nomor yang merupakan ID dari daftar dan jika wildcard tidak ditambahkan, secara default akan menjadi 0.0.0.0 atau /32.

### Tugas 3

Q Buatlah topologi berikut di Cisco Packet Tracer.



Dengan gateway default pada kedua perangkat yang diatur menuju interface router di masing-masing jaringan.

- Terapkan ACL masuk pada interface 192.168.1.254 Router0 tanpa aturan apa pun.
- Kemudian tambahkan satu aturan yang menolak 192.168.1.1/0 pada ACL yang sama.

Tugas: cantumkan perbedaan saat melakukan ping ke Server0 dari PC0 tanpa aturan apa pun dan dengan aturan yang menolak ditambahkan.

A **Aturan 1**

```
C:\>ping 192.168.2.1

Pinging 192.168.2.1 with 32 bytes of data:

Request timed out.
Reply from 192.168.2.1: bytes=32 time<1ms TTL=127
Reply from 192.168.2.1: bytes=32 time<1ms TTL=127
Reply from 192.168.2.1: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.2.1:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

**Aturan 2**

```
C:\>ping 192.168.2.1

Pinging 192.168.2.1 with 32 bytes of data:

Reply from 192.168.1.254: Destination host unreachable.
Reply from 192.168.1.254: Destination host unreachable.
Reply from 192.168.1.254: Destination host unreachable.
Reply from 192.168.1.254: Destination host unreachable.

Ping statistics for 192.168.2.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```



Di sisi lain, entri ACL IP yang diperluas yang sederhana dapat ditambahkan di mode konfigurasi global dengan menjalankan perintah sebagai berikut.

```
RouterA(config)# access-list <number> {permit | deny} <Source> <Destination>
```

Extended IP ACL jauh lebih rinci dibandingkan dengan standard IP ACL dengan konfigurasi detail yang mungkin dilakukan dengan menentukan port sumber/tujuan dan/atau protokol yang digunakan sehingga bisa sangat panjang. Contohnya, kita bisa mengizinkan host di 172.16.1.0/24 untuk menggunakan port sumber TCP lebih besar dari 9999 untuk mengakses semua port TCP pada server 4.4.4.4/32 kecuali port 23 pada nomor ACL 100 dengan memasukkan perintah berikut.

```
RouterA(config)# access-list 100 permit tcp 172.16.1.0 0.0.0.255 gt 9999  
host 4.4.4.4 neq 23
```

Internet Assigned Numbers Authority (IANA) bertanggung jawab untuk memelihara penetapan resmi nomor port untuk penggunaan tertentu. **Tabel resmi untuk port default dapat diakses di [sini](#)**. Dalam praktiknya, port yang digunakan untuk aplikasi dan layanan mungkin bervariasi dan dapat digunakan secara bergantian. Penggunaan port default bahkan tidak dianjurkan dalam konteks keamanan jaringan.

Daftar akses kemudian dapat diterapkan pada suatu interface dengan menggunakan perintah berikut, di mana 'in' menyaring paket yang diterima pada interface dan 'out' menyaring paket yang dikirim dari interface.

```
RouterA(config-if)# ip access-group 1 {in | out}
```

Standard dan extended IP access lists dibedakan melalui nomor ID yang digunakan selama pembuatan dengan rentang berikut. Penggunaan nama string sebagai pengganti nomor juga dimungkinkan.

Type	Range
Standard IP	1-99 & 1300-1999
Extended IP	100-199 & 2000-2699

Cara yang lebih baik untuk menambahkan ACL adalah dengan memasuki mode konfigurasi daftar kontrol akses jaringan saat membuat daftar dan menambahkan entri individual di

dalamnya. Memasuki mode konfigurasi daftar kontrol akses jaringan dapat dilakukan dengan menggunakan perintah ini.

```
RouterA(config)# ip access-list {standard | extended} {number | name}  
RouterA(config-{std | ext}-nacl)# {sequence number} {permit | deny} <Source>  
<Destination>
```

Mode konfigurasi daftar akses jaringan lebih baik karena memungkinkan spesifikasi nomor urutan untuk prioritas dan penghapusan entri daftar individual.

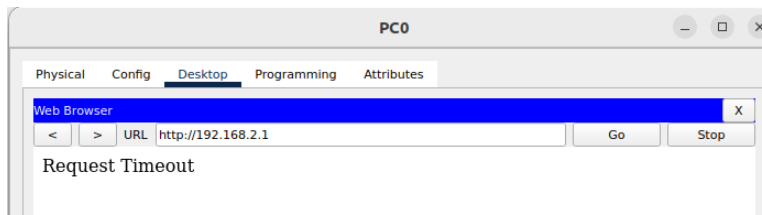
#### Tugas 4

- Q Menggunakan topologi dari tugas sebelumnya
- Hapus ACL yang ditambahkan pada tugas sebelumnya.
  - Buat daftar kontrol akses yang diperluas dengan nama atau nomor yang sembarangan.
  - Terapkan ACL keluar pada interface 192.168.2.254 Router0 tanpa aturan apa pun.
  - Tambahkan entri yang menolak akses TCP dari PC0 ke port 80 Server0.
  - Tambahkan entri yang mengizinkan akses TCP dari PC0 ke port 80 Server0 dengan nomor urutan yang lebih kecil.

Tugas: Cantumkan hasil saat melakukan ping dan mengakses web Server0 dari PC0 sebelum dan setelah menambahkan aturan kedua.

A **Aturan** [Tambahkan entri yang menolak akses TCP dari PC0 ke port 80 Server0.]

```
C:\>ping 192.168.2.1  
  
Pinging 192.168.2.1 with 32 bytes of data:  
  
Request timed out.  
Reply from 192.168.2.1: bytes=32 time<1ms TTL=127  
Reply from 192.168.2.1: bytes=32 time<1ms TTL=127  
Reply from 192.168.2.1: bytes=32 time<1ms TTL=127  
  
Ping statistics for 192.168.2.1:  
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```



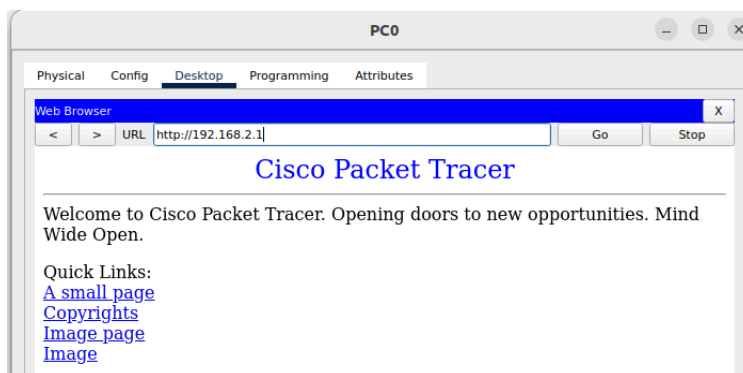
**Aturan** [Tambahkan entri yang mengizinkan akses TCP dari PC0 ke port 80 Server0 dengan nomor urutan yang lebih kecil.]

```
C:\>ping 192.168.2.1

Pinging 192.168.2.1 with 32 bytes of data:

Reply from 192.168.2.1: bytes=32 time<1ms TTL=127
Reply from 192.168.2.1: bytes=32 time<1ms TTL=127
Reply from 192.168.2.1: bytes=32 time<1ms TTL=127
Reply from 192.168.2.1: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.2.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```



Selain itu, daftar kontrol akses IPv6 juga dapat ditambahkan dengan cara yang serupa dengan memasukkan `ipv6` sebagai pengganti `ip` dalam perintah dengan satu tipe ACL yang tersedia di IPv6 mirip dengan extended IP ACL pada IPv4 dan hanya menggunakan nama string sebagai pengenal.

```
RouterA(config)# ipv6 access-group {name}
```

Apa itu IPv6? Pertanyaan bagus, silakan lanjut ke bagian berikutnya.

# IPv6

Internet Protocol versi 6 (IPv6) adalah versi baru dari pengalamatan IP (dan saat ini merupakan yang terbaru). Ini memperluas bit alamat dari 32 bit menjadi 128 bit, yang menyediakan jauh lebih banyak alamat dibandingkan pendahulunya (IPv4), lebih dari cukup untuk memberikan alamat unik bagi setiap perangkat yang terhubung ke jaringan yang ada. IPv6 bekerja dengan cara yang mirip dengan IPv4, dengan kemampuan yang lebih luas.

## Address Format

Alamat IPv6 direpresentasikan sebagai serangkaian bidang heksadesimal 16-bit yang dipisahkan oleh titik dua (:) dalam format `x:x:x:x:x:x:x:x`. Karena ruang alamat yang besar dan urutan yang panjang, alamat IPv6 dapat dipadatkan dengan mengompresi bidang heksadesimal nol yang berurutan (bagian 16-bit yang hanya berisi 0) menggunakan double colon (::). Kompresi dapat dilakukan di awal, tengah, atau akhir alamat. Misalnya, alamat loopback `0:0:0:0:0:0:0:1` dapat dipadatkan menjadi `::1`, atau alamat yang tidak ditentukan `0:0:0:0:0:0:0:0` dapat dipadatkan menjadi `::`. **Perlu dicatat bahwa Anda tidak dapat menggunakan double colon lebih dari sekali dalam satu alamat IPv6 (mengapa?).**

Mirip dengan subnet IPv4, IPv6 menggunakan prefix untuk merepresentasikan blok-blok yang berurutan secara bit dari seluruh ruang alamat. Karena prefix dapat menjadi cukup panjang dan oleh karena itu sulit untuk diekspresikan seperti subnet mask di IPv4, **IPv6 menggunakan format `ipv6-prefix/prefix-length`**. Di mana panjang prefix menentukan berapa banyak bit yang berurutan dari tinggi yang merupakan prefix (bagian jaringan dari alamat, sama seperti notasi CIDR di IPv4).

Tugas 5	
Q	<p>Unduhlah file ini <a href="#">IPv6-start.pkt</a>⬇️. Mulailah dengan menetapkan alamat IPv6 ke setiap perangkat seperti yang tercatat dalam file packet tracer.</p> <p>Penetapan alamat IPv6 pada PC dan Server dapat dilakukan melalui aplikasi alamat IP, menggunakan kolom IPv6 di bawah kolom IPv4. Pastikan untuk mengisi default gateway dan server DNS dengan benar (gunakan server "DNS Server" sebagai server DNS untuk semua perangkat).</p>

Untuk mengonfigurasi alamat IPv6 pada router, konfigurasi di mode interface dengan cara yang mirip dengan mengonfigurasi alamat IPv4 di modul sebelumnya:

```
Router(config)#interface {interface_id}
Router(config-if)#ipv6 enable
Router(config-if)#ipv6 address {ipv6_address}/{prefix_length}
Router(config-if)#no shutdown
```

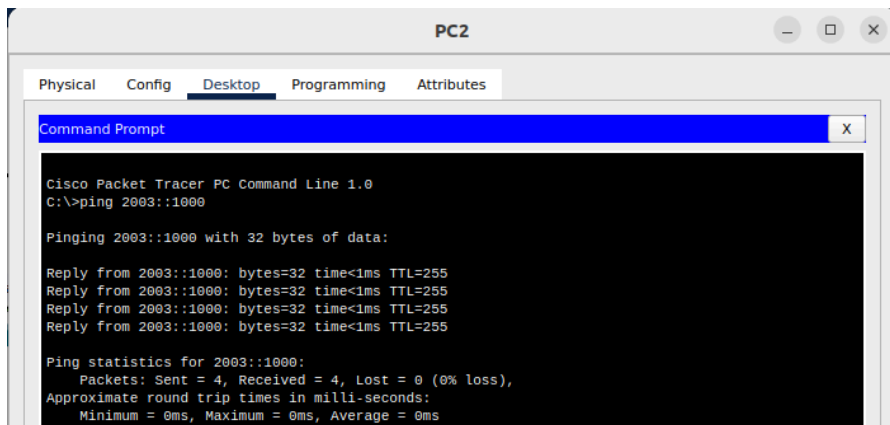
Hint: untuk menampilkan alamat IPv6 di interface, gunakan:

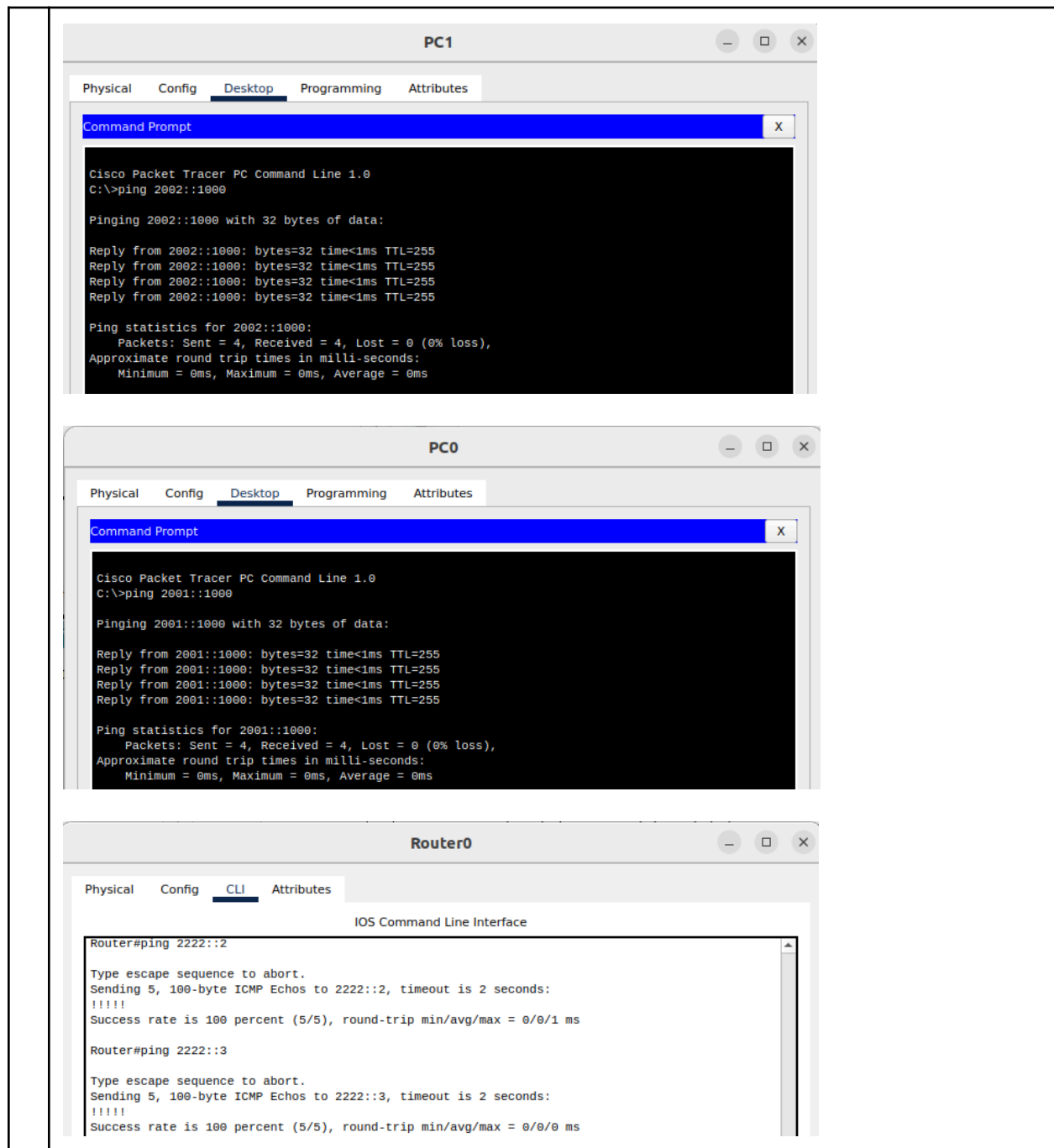
```
show ipv6 int brief
```

Simpan progress Anda saat ini ke file baru, untuk memudahkan pengaturan ulang konfigurasi routing untuk tugas berikutnya.

Tugas: Cobalah untuk melakukan ping ke router yang berdekatan dari setiap PC (router yang terhubung langsung ke PC), dan lakukan ping ke setiap router lainnya dari Router0, dan tampilkan hasilnya!

A





## Addressing and Routing

Unicast IPv6 routing dilakukan dengan cara yang sama seperti routing IPv4. Kita telah membahas beberapa dynamic routing protocols dalam aktivitas laboratorium sebelumnya, dan kita hanya akan menggunakan RIP untuk dynamic routing dalam tugas ini.

## Tugas 6

Q Melanjutkan dari tugas sebelumnya, kita akan mengonfigurasi routing statik IPv6 di jaringan.

Untuk memulai, kita perlu mengaktifkan routing unicast di router.

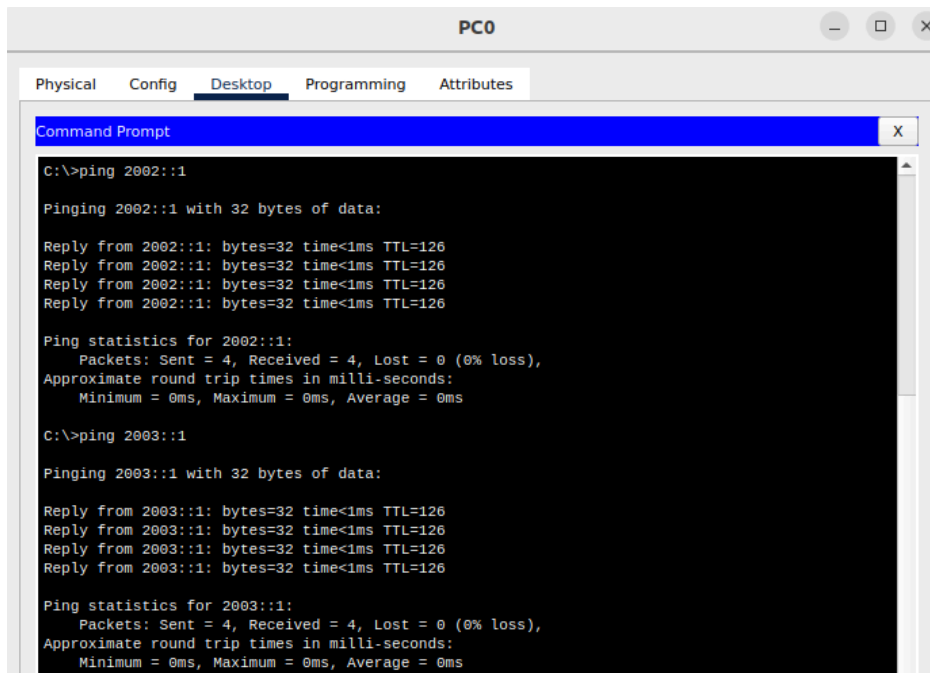
```
Router(config)#ipv6 unicast-routing
```

Kemudian, daftarkan routing statis dengan cara yang sama seperti routing statik IPv4 menggunakan

```
ipv6 route {destination_network} {next_hop}
```

Tugas: setelah mengonfigurasi routing di ketiga router, coba ping semua PC dari PC0, dan coba akses `itb.ac.id` dari PC1 (*don't worry*, DNS record sudah dikonfigurasi). Tunjukkan hasilnya!

A



```
PC0
Physical Config Desktop Programming Attributes
Command Prompt
C:\>ping 2002::1

Pinging 2002::1 with 32 bytes of data:

Reply from 2002::1: bytes=32 time<1ms TTL=126
Reply from 2002::1: bytes=32 time<1ms TTL=126
Reply from 2002::1: bytes=32 time<1ms TTL=126
Reply from 2002::1: bytes=32 time<1ms TTL=126

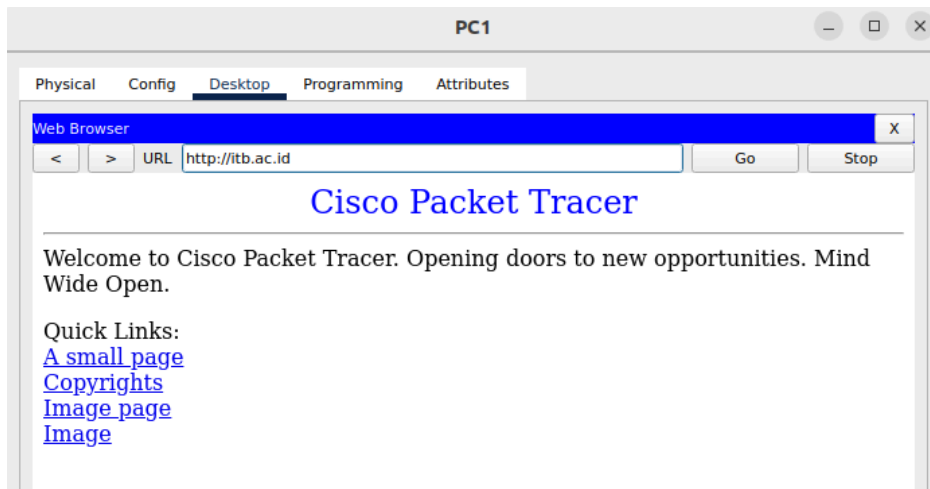
Ping statistics for 2002::1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 2003::1

Pinging 2003::1 with 32 bytes of data:

Reply from 2003::1: bytes=32 time<1ms TTL=126
Reply from 2003::1: bytes=32 time<1ms TTL=126
Reply from 2003::1: bytes=32 time<1ms TTL=126
Reply from 2003::1: bytes=32 time<1ms TTL=126

Ping statistics for 2003::1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```



Q Setelah mencoba routing statik IPv6, sekarang saatnya mencoba routing dinamis. Sebelum melanjutkan, bersihkan semua konfigurasi routing statis (Anda dapat menggunakan "Power Cycle Devices" (restart semua perangkat) dengan `alt+s`) atau restart Packet Tracer tanpa *save* jika Anda tidak sengaja *save running-config* ke *startup-config*).

Mulailah dengan menginisialisasi protokol RIP menggunakan

```
ipv6 router rip {rip_name}
```

Konfigurasi setiap interface router yang tergabung pada *routing protocol* (**yaitu semua interface**) untuk mengaktifkan routing RIP dengan perintah

```
ipv6 rip {rip_name} enable
```

..selesai! Sekarang konfigurasi semua *interface* router yang menghubungkan masing-masing jaringan dengan langkah-langkah konfigurasi yang sama untuk menyelesaikan proses ini.

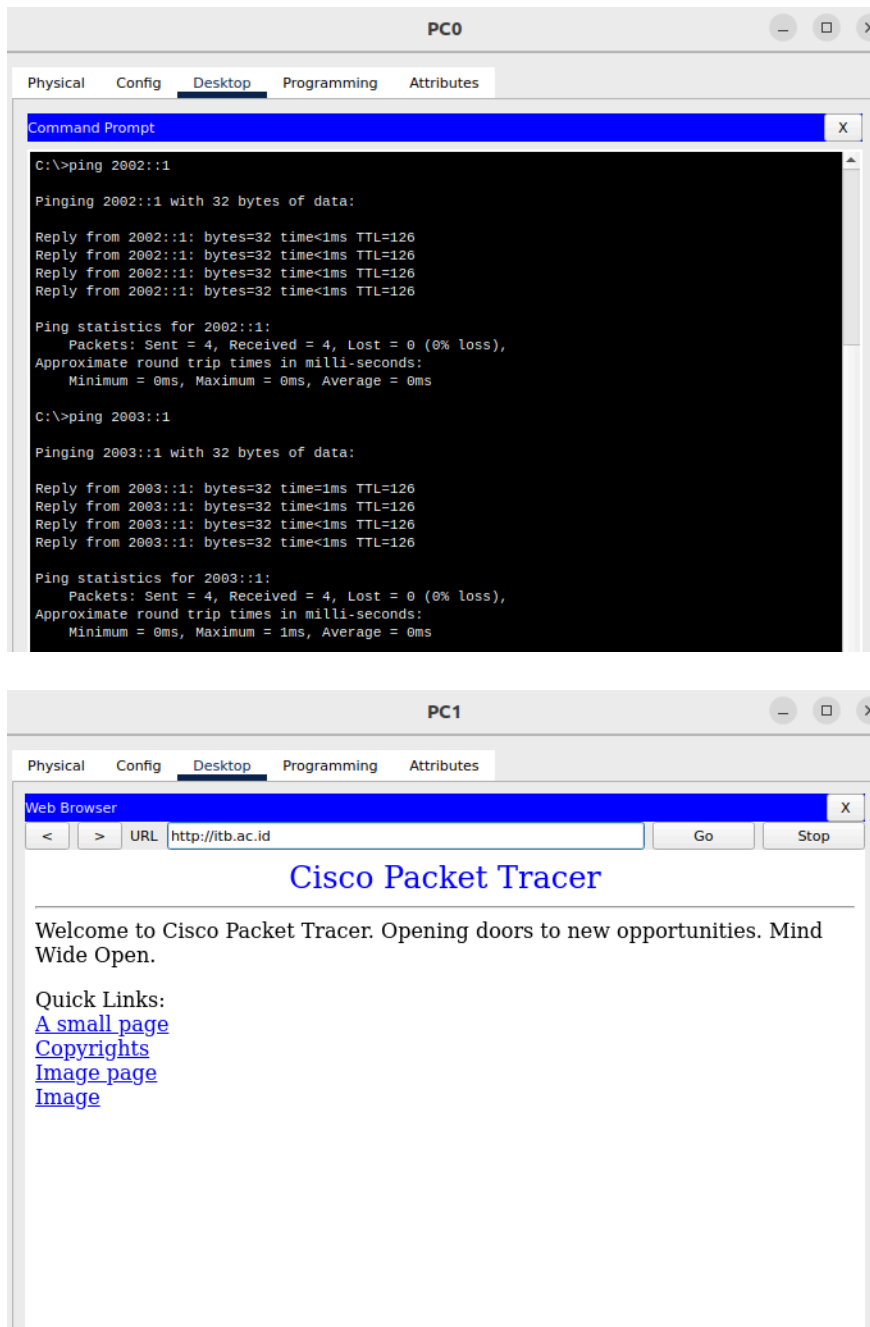
*Hint:* untuk menampilkan tabel routing IPv6, gunakan

```
show ipv6 route
```

Tugas: setelah mengonfigurasi routing di ketiga router, coba lakukan ping ke semua PC dari PC0, dan coba akses `itb.ac.id` dari PC1 lagi dan tampilkan hasilnya!



A



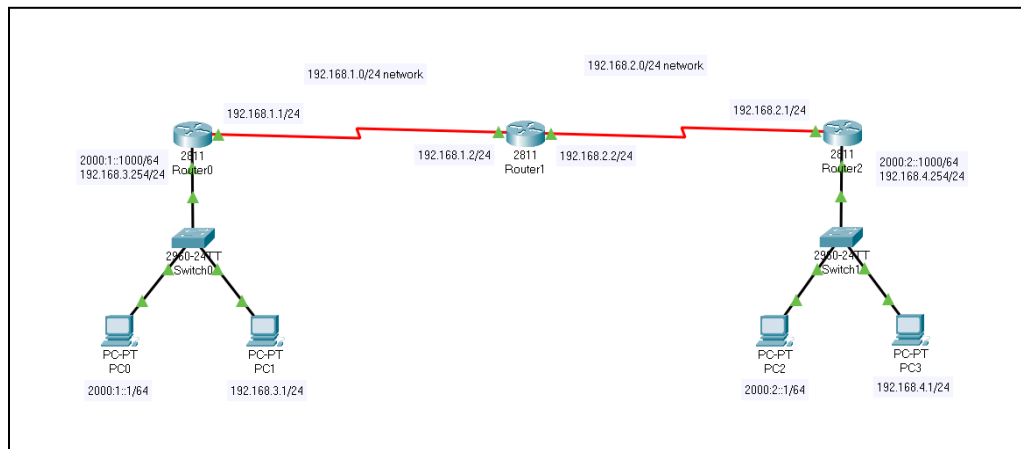
## IPv6 Tunneling over IPv4

Implementasi IPv6 memerlukan *effort* yang lumayan untuk diterapkan pada infrastruktur jaringan saat ini, dan kedua skema pengalamatan tidak kompatibel satu sama lain. Jadi, implementasi IPv6 hanya dapat dilakukan dengan *rolling deployment*. Untuk menjaga agar jaringan tetap berjalan di IPv4 dan tetap terhubung dengan jaringan yang sudah berjalan di

IPv6, kita perlu cara untuk menjembatani kedua skema pengalamatan dan routing-nya. Solusi yang banyak digunakan adalah IPv6 tunneling over IPv4.

## Tugas 7

Q Unduh file ini [IPv6-tunnel-start.pkt](#) ↓.



Dalam file tersebut, terdapat infrastruktur jaringan IPv4 yang sudah berjalan, dengan PC (PC0 & PC2) yang menggunakan IPv6 ditambahkan di setiap jaringan akhir. Semua perangkat telah dikonfigurasi dengan benar, tetapi router hanya dikonfigurasi untuk jaringan IPv4. Sebelum melanjutkan, pastikan jaringan berfungsi dengan baik dengan ping dari PC1 ke PC3.

Sekarang, kita ingin menghubungkan PC0 ke PC2 dengan infrastruktur jaringan yang sudah ada tanpa merusak jaringan untuk PC1 & PC3. Untuk melakukan ini, kita perlu mengonfigurasi IPv6 & tunneling IPv6 melalui IPv4 di router.

Untuk memulai, kita perlu mengaktifkan IPv6 & routing unicast di Router0 & Router2

```
Router(config)#ipv6 unicast-routing
Router(config)#int {interface_id}
Router(config-if)#ipv6 address {IPv6_address}
Router(config-if)#ipv6 enable
Router(config-if)#exit
```

Setelah mengonfigurasi alamat IPv6, kita perlu mengonfigurasi interface tunnel dan jaringannya. Mulailah dengan mengonfigurasi interface tunnel.

```
Router(config)#interface Tunnel0
Router(config-if)#ip address {tunnel_address}{subnet_mask}
Router(config-if)#ipv6 address {tunnel_ipv6_address}
Router(config-if)#ipv6 enable
Router(config-if)#tunnel source Serial0/0/0
Router(config-if)#tunnel destination {target_router_address}
Router(config-if)#tunnel mode ipv6ip
Router(config-if)#exit
```

Anda dapat menggunakan konfigurasi berikut, atau menggunakan nilai apa pun yang ingin Anda coba:

- Gunakan 172.16.0.0/16 sebagai jaringan interface tunnel (misalnya dengan alamat 172.16.0.1/16 untuk Router0 & 172.16.0.2/16 untuk Router2).
- Gunakan 2000::/64 sebagai prefix dari interface tunnel (misalnya dengan alamat 2000::1/64 untuk Router0 & 2000::2/64 untuk Router2).

Sekarang interface tunnel telah disetel di kedua perangkat, bagaimana dengan routing?

Paket IPv4 (yang membungkus paket IPv6) sudah langsung diteruskan oleh alamat "tunnel source" dengan tujuan ke alamat "tunnel destination", yang telah dirutekan di atas routing IPv4 yang telah diatur sebelumnya. Dengan memeriksa field protokol dalam header IP (di mana 41 menandakan protokol enkapsulasi IPv6), paket kemudian akan dikirim ke interface tunnel.

Bagaimana dengan routing IPv6? Sekarang kita perlu mengonfigurasi routing IPv6. Kita dapat menggunakan metode routing IPv6 apa pun, tetapi kita akan menggunakan RIP untuk tugas ini.

Pertama, aktifkan protokol IPv6 RIP dengan

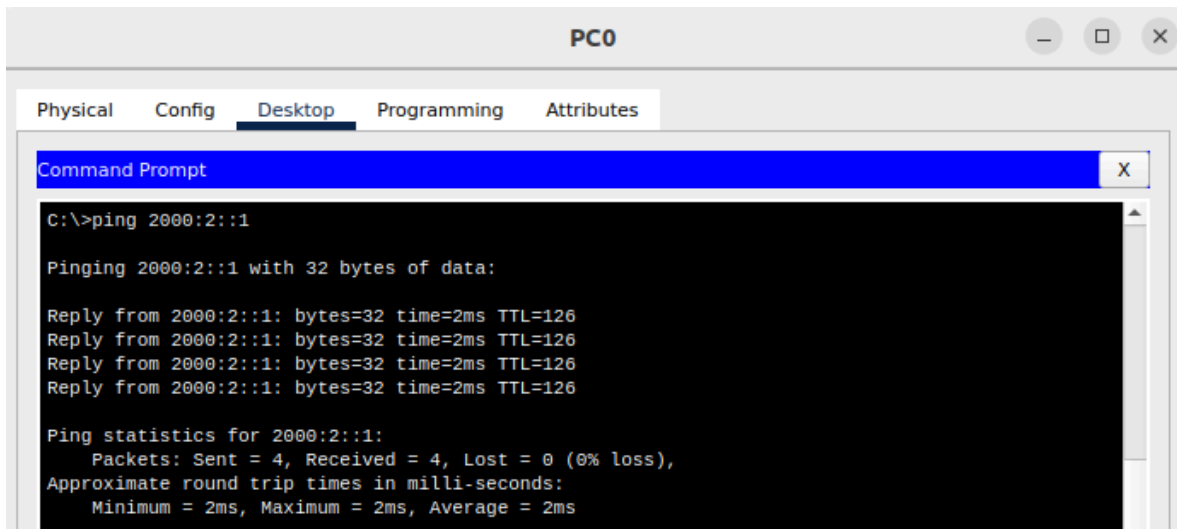
```
ipv6 router rip {rip_name}
```

Kemudian konfigurasikan setiap interface (yaitu interface Router yang terhubung ke PC, dan interface tunnel) untuk berpartisipasi dalam protokol RIP yang sama dengan

```
ipv6 rip {rip_name} enable
```

Tugas: Setelah mengonfigurasi tunneling di kedua router, tampilkan hasil ping dari PC0 ke PC2! Jelaskan proses enkapsulasi/dekapsulasi paket dan proses routing!

A



The screenshot shows a window titled "PC0" with tabs for Physical, Config, Desktop, Programming, and Attributes. The Desktop tab is active, displaying a Command Prompt window. The Command Prompt shows the command "C:\>ping 2000:2::1" and its output:

```
C:\>ping 2000:2::1

Pinging 2000:2::1 with 32 bytes of data:

Reply from 2000:2::1: bytes=32 time=2ms TTL=126
Reply from 2000:2::1: bytes=32 time=2ms TTL=126
Reply from 2000:2::1: bytes=32 time=2ms TTL=126
Reply from 2000:2::1: bytes=32 time=2ms TTL=126

Ping statistics for 2000:2::1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 2ms, Average = 2ms
```

1. PC0 mengirimkan ICMP Echo Request ke PC2 dengan dest IPv6 2000:2::11
2. Router0 menerima ICMP Echo Request dan memasukkannya ke Tunnel 10 lalu dienkapsulasi dalam IPv4 dengan field protocol diset ke 41.
3. Paket yang sudah dienkapsulasi dikirimkan ke Router2 dengan destination IPv4 192.168.1.2
4. Router2 menerima paket dan melihat bahwa field protocol pada IP Header adalah 41, maka IPv4 header dicabut. Selanjutnya paket diteruskan ke PC2 dengan tujuan IPv6 2000:2::1
5. Paket diterima oleh PC2, lalu dikirimkan kembali ICMP Echo Reply dengan dest IPv6 2000:1::1.
6. Proses yang sama dilakukan seperti sebelumnya

## TIPS

1. Untuk praktikum 4, *cheat sheet* akan disiapkan juga sehingga pelajarilah **alur** dan **makna** dari setiap langkah yang dilakukan alih-alih menghafalkan sintaks.
2. Ketika menambahkan entri di *access list*, gunakan *sequence number* yang *sparse* supaya mudah ketika ingin menambahkan sebuah entri di antara dua entri yang sudah ada.
  - a. Contoh: misal kita sudah menambahkan 2 entri dengan sequence number berturut-turut 1 dan 2. Sekarang, kita ingin menambahkan sebuah entri di antara 2 entri ini, sulit bukan? Kita harus mengubah *sequence number* entri 2 menjadi 3 terlebih dahulu.
  - b. Lebih baik ketika menambahkan 2 entri yang awal, kita gunakan *sequence number* (misalnya) 10 dan 20 sehingga ada 9 slot entri antara kedua entri ini yang bisa kita gunakan (11-19).

# Referensi

Cisco. (n.d.). *Cisco Networking Academy*. <https://www.netacad.com>

Lammle, T. (2020). *CCNA certification study guide: Exam 200-301*. Sybex.

[Cheat Sheet](#) (akan diperbolehkan untuk dibuka saat praktikum 4).

DNS A record vs. NS record? [DNS A vs NS record - Server Fault](#)

Bagaimana *DNS resolution* bekerja? [What is DNS? | How DNS works | Cloudflare](#)