

ALGORITMA PERANCANGAN PIRANTI LUNAK APLIKASI NETMAP UNTUK PEMETAAN TOPOLOGI JARINGAN KOMPUTER

Fransiscus Ati Halim, Meilinie

Universitas Pelita Harapan, Indonesia

UPH Tower, Lippo Karawaci, Tangerang 15811, Indonesia

E-mail: fransiscus_halim@uph.edu

ABSTRACT

Two common ways for tackling network problems are troubleshooting or upgrading. For the success of both processes, a complete documentation, which encounters a number of essential factors, should be in existence. Among of these known factors is network topology.

When a network is to cover a large geographical area, which is a common case nowadays, the topology mapping is likely to be more difficult. Motivated by this reason, a specific application, called Netmap, was designed to be able to map any given network topology remotely. Through the testing phase on various network topologies, Netmap showed a success percentage of 88% with an average execution time of 32.05 seconds. Besides, Netmap can provide the user with specific information the about existing devices, such as hostname, IP address and MAC address. For further improvement, the future application is expected to be able to deal with extracting more data and compatible on multiplatform.

Kata kunci: network topology algorithm

1. PENDAHULUAN

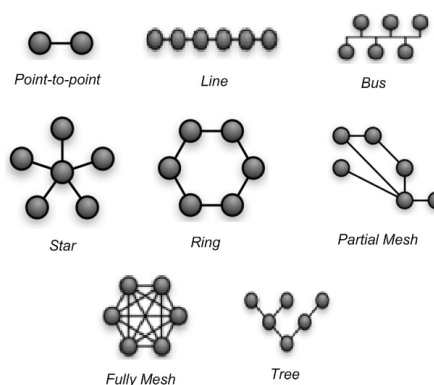
Selain jaringan komputer dibangun sesuai kebutuhan pengguna, dokumentasi yang lengkap tentang jaringan komputer juga dibutuhkan oleh administrator jaringan komputer dan pihak-pihak tertentu dalam suatu organisasi atau perusahaan. Dalam merancang aplikasi pemetaan topologi jaringan sebagai salah satu bentuk dokumentasi, permasalahan pada aplikasi Netmap yang utama ialah bagaimana cara mendapat data yang dibutuhkan dari jaringan yang terhubung dengan host.

Batasan masalah dari perancangan aplikasi piranti lunak untuk pemetaan topologi jaringan komputer ialah aplikasi dirancang untuk dapat berjalan efektif pada spesifikasi komputer; Sistem Operasi Microsoft Windows XP Professional, Intel(R) Celeron(R) M CPU 1,86 GHz dan RAM 1,5GB. Pemetaan dilakukan aplikasi untuk jaringan dengan media kabel (LAN atau WAN) dan lebih cocok digunakan pada topologi jaringan *point-to-point*, *line*, *star*, dan *tree*. Jalannya aplikasi tidak memperhitungkan kecepatan atau *bandwidth* jaringan dan tidak bersifat *real-time*. Topologi jaringan dapat diperoleh bila port 23 untuk Telnet pada host yang terhubung dengan peralatan jaringan komputer lain tidak ditutup dan telah diketahui password administrator-nya. Penelitian yang dilakukan bersifat eksperimental di laboratorium. Oleh karena itu, pemrograman untuk koneksi Telnet dibuat untuk Cisco router dan Catalyst switch dengan konfigurasi *cisco discovery protocol* yang diaktifkan. Seluruh password yang digunakan pada jalur vty router dan switch sama, demikian pula dengan password untuk mengakses *privileged mode*.

2. LANDASAN TEORI

2.1 Konsep Topologi Jaringan Komputer

Topologi jaringan merupakan pembelajaran dari penyusunan atau pemetaan elements antara lain jalur, nodes. pada jaringan, terutama pada interkoneksi fisik (*real*) dan logika (*virtual*) antar node[8]. Secara umum, topologi fisik dibagi dalam beberapa tipe yaitu *point-to-point*, *bus* (*linear* dan *distributed bus*), *star* (*extended* dan *distributed star*), *ring*, *mesh* (*fully* dan *partially connected*), dan *tree* (*hierarchical*). Berbagai bentuk topologi fisik ditunjukkan pada gambar 1.



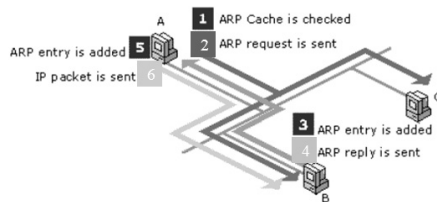
Gambar 1 Berbagai bentuk topologi fisik

Sumber : Wikipedia, 2007

2.2 Address Resolution Protocol (ARP)

Proses resolusi pengalamatan fisik terjadi saat IP (*Internet Protocol*) mengirim informasi untuk ARP, kemudian ARP menerima paket IP,

forwarding IP address, dan interface yang digunakan untuk meneruskan paket. Baik performa pengiriman langsung atau tidak langsung, ARP mengikuti proses yang digambarkan pada gambar 2

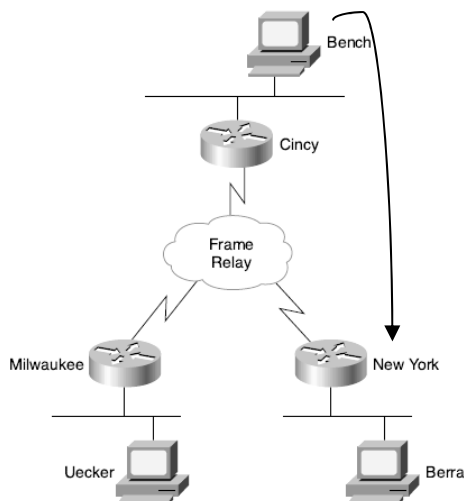


Gambar 2. Proses ARP

Sumber : MSDN, 1998

2.3 Telnet

Telnet merupakan protokol emulator terminal yang termasuk dalam bagian protokol TCP/IP dan mendukung layanan *remote terminal-connection*. Pada gambar 3. ditunjukkan peristiwa Telnet yang dilakukan dari komputer Bench ke *router* New York melewati jaringan *Frame Relay*.



Gambar 3. Peristiwa Telnet untuk konfigurasi *router*

Sumber: CCNA Intro Exam Certification Guide, 2004

3. METODE PENGUMPULAN DATA

Pembuatan aplikasi Netmap untuk memetakan hubungan antar peralatan jaringan yang sudah ada menjadi sebuah topologi diawali dengan pengumpulan data. Data dari *host* dan setiap peralatan yang ada pada jaringan diperoleh dengan penggunaan beberapa kepastakaan dan protokol. Kepustakaan dan protokol yang digunakan untuk pengumpulan data yaitu kepastakaan WinPcap, IP Helper, Protokol ARP, dan Telnet, serta kumpulan fungsi yang sejenis untuk mengakses fungsi-fungsi internal pada sistem operasi dalam API (*Application Programming Interfaces*) atau *framework* seperti komponen *BackgroundWorker*. *BackgroundWorker*

merupakan komponen yang terdapat pada .NET *framework* dan digunakan sebagai implementasi dari *multithread*. *Multithreading* merupakan konsep untuk menjalankan beberapa *thread* atau proses secara bersamaan. Secara *default*, aplikasi Windows menggunakan *single thread of execution*, tetapi dengan penggunaan *BackgroundWorker*, dapat ditambahkan *thread* lagi sehingga proses-proses pada aplikasi .NET dapat dieksekusi secara independen.

3.1 Analisis Permasalahan

Jaringan komputer saat ini tidak hanya digunakan untuk berbagi data dan *resources* saja, tetapi penggunaan komputer telah meluas untuk kepentingan komunikasi suara dan multimedia melalui *Local Area Network* (LAN) maupun *Wide Area Network* (WAN).

Seiring dengan perkembangan jaringan komputer, dibutuhkan pula dokumentasi yang lengkap tentang jaringan komputer tersebut. Hal ini penting untuk mengatasi masalah yang terjadi pada jaringan misalnya perubahan konfigurasi, penambahan workstation sesuai dengan kebutuhan jangka panjang. Oleh karena itu, perlu dibuat sebuah piranti lunak aplikasi untuk memetakan jaringan komputer.

3.2 Peran WinPcap

Kepustakaan WinPcap digunakan untuk mengambil data dari *host* sebagai komputer tempat dieksekusinya aplikasi piranti lunak. Data tentang *host* yang perlu diketahui berhubungan dengan *interface* yang dimiliki. Data pada *interfaces* komputer yang diambil meliputi nama, deskripsi, *loopback*, IP address, *subnet mask*, dan *broadcast address*. Tampilan data-data tersebut dapat dilihat pada gambar 4 di bawah ini.

```

C:\Documents\Semester 7\Skrpsi\Riset\WinPcap\WpdPack_4_0_1\WpdPack\Examples-remote\listDebug\list.exe
rpcap://\Device\NPF_{GenericDialupAdapter}
Description: Network adapter 'Adapter for generic dialup and UPN capture'
on local host
Loopback: no
rpcap://\Device\NPF_{1EAB651C-E7EC-4921-R41A-C2E335B5C89D}
Description: Network adapter 'Realtek RTL8169/8110 Family Gigabit Ethernet NIC'
on local host
Loopback: no
Address Family: #2
Address Family Name: AF_INET
Address: 10.10.10.15
Netmask: 255.255.255.224
Broadcast Address: 255.255.255.255
rpcap://\Device\NPF_{8E41AD4D-5144-412a-8A95-1B523CD39E6A}
Description: Network adapter 'Intel(R) PRO/Wireless 3945ABG Network Connection'
on local host
Loopback: no
Address Family: #2
Address Family Name: AF_INET
Address: 10.10.10.4
Netmask: 255.255.255.224
Broadcast Address: 255.255.255.255
    
```

Gambar 4. Hasil *print screen* eksekusi fungsi WinPcap untuk menampilkan data *interfaces*.

3.3 Peran IP Helper

Internet Protocol Helper (IP Helper) API yang merupakan bagian dari kepastakaan MSDN digunakan untuk memperoleh data tentang IP address gateway dari setiap *interface* yang dimiliki

host. Selain itu, pengiriman *frame ARP request* dapat dilakukan dengan penggunaan fungsi pada IP *Helper*.

3.4 Penggunaan ARP

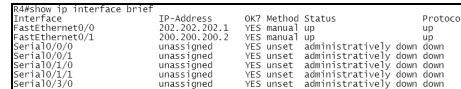
ARP dilakukan saat *host* terhubung dengan jaringan yang aktif. Dengan pengiriman *ARP request*, diperoleh data tentang keberadaan peralatan jaringan lain pada *network address* yang sama.

ARP request dikirim ke semua *IP address* pada jaringan kecuali *network address*, *broadcast address*, dan *IP address host*. Keberadaan peralatan jaringan lain dapat diketahui bila *reply* dari pengiriman *ARP request* diterima.

3.5 Penggunaan Telnet

Selain protokol ARP yang digunakan untuk mendapatkan data dari peralatan jaringan lain, pada aplikasi digunakan pula Telnet. Saat *host* terhubung dengan jaringan yang aktif, Telnet dilakukan pertama kali ke *IP address gateway*. Telnet dilakukan untuk mendapat akses langsung ke *gateway* yang juga merupakan *router*. Pada *router* dan *switch*, Telnet digunakan agar diperoleh data dari peralatan jaringan yang berbeda *network address* dengan *host*. Data-data yang diperoleh setelah Telnet dilakukan ialah nama *router* atau *switch* serta data pada masing-masing *interface* seperti nama, *IP address*, status dan protokol. Selain itu, diperoleh juga data peralatan jaringan lain yang terhubung langsung dengan *router* atau *switch*.


Nama *router* atau *switch* yang dikenal pula sebagai *device ID* diperoleh setelah akses Telnet ke *router* atau *switch* berhasil. Setelah *password* untuk jalur *vtty* dimasukkan, data berupa *string* yang diterima dimasukkan ke *buffer* dan disimpan mulai *string* ketiga sampai sepanjang *buffer* - 3 sebagai nama *router* atau *switch*. Data yang berhubungan dengan *interfaces router* diperoleh setelah dikirimkan perintah `show ip interface brief` seperti pada gambar 5 Tetapi, data *interface switch* yang merupakan *IP address* diperoleh setelah `show cdp entry <device ID>`.



```
Router#show ip interface brief
Interface          IP-Address      OK? Method Status    Protocol
FastEthernet0/0    202.202.202.1  YES manual up        up
FastEthernet0/1    200.200.200.2  YES manual up        up
Serial0/0/0        unassigned     YES unset  administratively down down
Serial0/0/1        unassigned     YES unset  administratively down down
Serial0/1/0        unassigned     YES unset  administratively down down
Serial0/1/1        unassigned     YES unset  administratively down down
Serial0/3/0        unassigned     YES unset  administratively down down
```

Gambar 5. Hasil *print-screen* eksekusi perintah `show ip interface brief`

Setelah pengambilan data *interfaces*, pada *router* atau *switch* juga diberikan perintah `show cdp neighbors`. Perintah ini digunakan untuk mengetahui peralatan jaringan lain yang berhubungan langsung dengan *router* atau *switch*. Selain itu, dengan perintah `show cdp neighbors` dapat diketahui *device ID* dan *compatibility*-nya. *Compatibility* digunakan untuk menentukan jenis peralatan jaringan, diantaranya “R” untuk *router* dan “S” untuk *switch*.



```
Router#show cdp neighbors
Capability Codes: (R) - Router, T - Trans Bridge, B - Source Route Bridge
                  (S) - Switch, H - Host, I - IGMP, r - Repeater

Device ID         Local Interface   Holdtime  Capability  Platform  Port ID
SW2960            Eth 0             150      S I         WS-C2960-2Fas 0/3
```

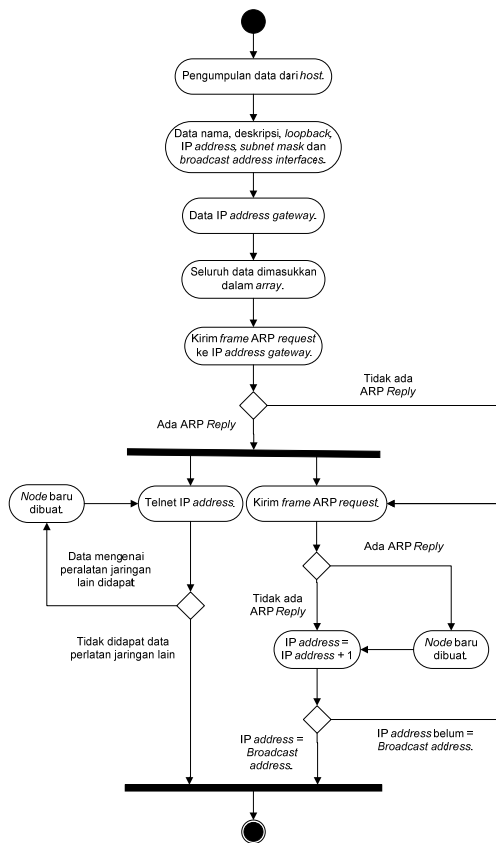
Gambar 6. Hasil *print-screen* eksekusi perintah `show cdp neighbors`

Telnet juga dilakukan pada setiap *device ID* yang terdapat dari hasil eksekusi perintah `show cdp neighbors` seperti gambar 6. *IP address* dari setiap *device ID* yang ada diperoleh dengan memasukkan perintah `show cdp entry <device ID>`. Setelah *IP address* diketahui, Telnet kemudian dilakukan agar dapat diperoleh data dari peralatan jaringan tersebut. Akan tetapi, sebelumnya dilakukan pengecekan apakah *IP address* tersebut merupakan *IP address* salah satu *interface* dari peralatan jaringan atau *node* yang telah ada. Hal ini digunakan untuk menghindari terjadinya perulangan Telnet.

4. PERANCANGAN

4.1 Pembuatan Algoritma Pemetaan Topologi Jaringan Komputer

Metode yang digunakan untuk pengumpulan data kemudian dirangkum menjadi algoritma. Hal ini bertujuan agar seluruh informasi tentang jaringan yang terhubung dengan *host* diperoleh. Algoritma pada piranti lunak aplikasi Netmap untuk pemetaan topologi jaringan komputer ditunjukkan dengan *activity diagram* pada gambar 7.



Gambar 7. Activity diagram algoritma piranti lunak aplikasi Netmap

4.2 Metode Pemetaan pada Aplikasi Netmap

Pendekatan yang digunakan dalam pemrograman aplikasi ialah *Object-Oriented Programming* (OOP). Dengan pendekatan ini, setiap peralatan jaringan diperlakukan sebagai *object*. *Object* utama pada aplikasi ialah *node* dan untuk menyimpan data sejumlah *node* digunakan *array arrnode*. *Node* digunakan untuk mewakili peralatan jaringan dan menyimpan berbagai data tentang *image*, struktur *interfaces*, *label* dan data koordinat pada *canvas*. Data *interfaces* peralatan jaringan dibuat dalam bentuk struktur *array arrinterface* didalam *node* yang terdiri atas Name, Description, MediaType, Loopback, macAddress, ipAddress, Netmask, netAddr, Broadcast, Status, dan Protocol.

Setelah data disimpan dalam struktur, secara bersamaan model topologi jaringan dibuat. Hal ini dilakukan dengan menggunakan *BackgroundWorker* pada program. Terdapat dua *worker* yang digunakan agar program dapat dieksekusi lebih cepat, yaitu pada saat melakukan pengiriman *ARP request* dan *Telnet*. Dengan penggunaan *worker*, penggambaran *node* tidak menunggu *ARP* atau *Telnet* selesai dilakukan, sehingga proses dapat dilakukan secara bersamaan

4.3 Evaluasi Hasil Pemetaan Topologi Jaringan Komputer

Beberapa hal yang dapat dievaluasi berdasarkan hasil pengujian piranti lunak aplikasi Netmap dalam melakukan pemetaan ialah mengenai persentase keberhasilan saat memetakan jaringan komputer dengan berbagai topologi dan penyebab kesalahan yang sering terjadi saat aplikasi berjalan. Persentase keberhasilan dan waktu eksekusi diperoleh dari pengujian berulang-ulang pada setiap topologi yang telah dibahas pada subbab 4.1. Berdasarkan persentase keberhasilan dan rata-rata waktu eksekusi yang didapat dari pengujian seluruh topologi, diperoleh persentase dan waktu eksekusi rata-rata yang ditunjukkan pada tabel 1.

Tabel 1. Perhitungan % & waktu eksekusi rata-rata.

Topologi	Persentase Keberhasilan (%)	Rata-rata Waktu Eksekusi (detik)
<i>Point-to-point</i>	100	10,4
<i>Line</i>	80	64,25
<i>Star 1</i>	80	6,75
<i>Star 2</i>	80	41,25
<i>Tree</i>	100	37,6
Rata-rata	88	32,05

Seperti ditunjukkan pada tabel 1, persentase keberhasilan untuk setiap topologi berbeda-beda dan topologi yang dihasilkan dari eksekusi aplikasi tidak selalu tepat. Kegagalan eksekusi aplikasi sering terjadi saat proses *Telnet* dilakukan. Terkadang, waktu penerimaan *string* oleh *host* tidak tepat sehingga menyebabkan *buffer* yang disediakan untuk memperoleh data nama *router* atau *switch* kosong. Berdasarkan penelitian yang dilakukan, kegagalan saat menerima *string* terkait dengan pertukaran *CDP* dan *hello packet* pada peralatan jaringan Cisco.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Beberapa hal yang dapat disimpulkan setelah melakukan penelitian ialah :

1. Konsep *multithread* berhasil diterapkan dengan penggunaan *BackgroundWorker* sehingga aplikasi berjalan responsif. Berdasarkan pengujian yang telah dilakukan, waktu eksekusi rata-rata dari awal dijalankannya aplikasi sampai menampilkan topologi ialah 32,05 detik.
2. Aplikasi tidak hanya menampilkan topologi jaringan komputer saja, tetapi juga informasi

tambahan berupa nama *host* pada komputer tempat dijalankannya aplikasi, *IP address* pada seluruh *node* dan *MAC address* pada *network address* yang sama dengan *host*.

5.2 Saran

Saat ini, akses langsung ke peralatan jaringan dilakukan melalui Telnet. Pada pengembangan selanjutnya dapat digunakan perintah lain untuk mengakses data pada *router* atau *switch* Cisco. Agar aplikasi tidak berjalan statis, dapat digunakan *dialog box* untuk memasukkan *password* jalur *vty* dan *privileged mode*.

Selain diperolehnya topologi jaringan komputer, Algoritma Netmap dapat dikembangkan untuk mengambil data-data seperti lalu lintas jaringan, keberadaan serta kesibukan prosesor komputer lain pada jaringan dan sebagainya. Dengan penelitian lebih lanjut, diharapkan aplikasi juga dapat digunakan untuk memetakan topologi jaringan komputer *multiplatform* baik dalam hal sistem operasi maupun dari berbagai vendor peralatan jaringan.

PUSTAKA

Cisco Inc. *Cisco Networking Academy Program*. 2003

Nance, Barry. *Introduction to Networking*. Indianapolis: Que Corporation, 1997.

<http://www.knowdotnet.com/articles/backgroundworker.html>, 5 Oktober 2007.

<http://www.winpcap.org/archive/4.01-WpdPack.zip>, 26 Juli 2007.

Microsoft Corporation. *IP Helper*. Maret 2005. ms-help://MS.MSDNQTR.v80.en/MS.MSDN.v80/MS.WIN32COM.v10.en/iphlp/iphlp/ip_helper_start_page.htm, 14 September 2007.

"Introduction to TCP/IP." September 1998. <ms-help://MS.MSDNQTR.v80.en/MS.MSDN.v80/MS.WIN32COM.v10.en/dniph/html/tcpipintro.htm>, 8 Oktober 2007.

Odom, Wendell. *CCNA Intro Exam Certification Guide*. Indianapolis: Cisco Press, 2004.

Network topology. http://en.wikipedia.org/wiki/Network_topology, 10 Desember 2007.

