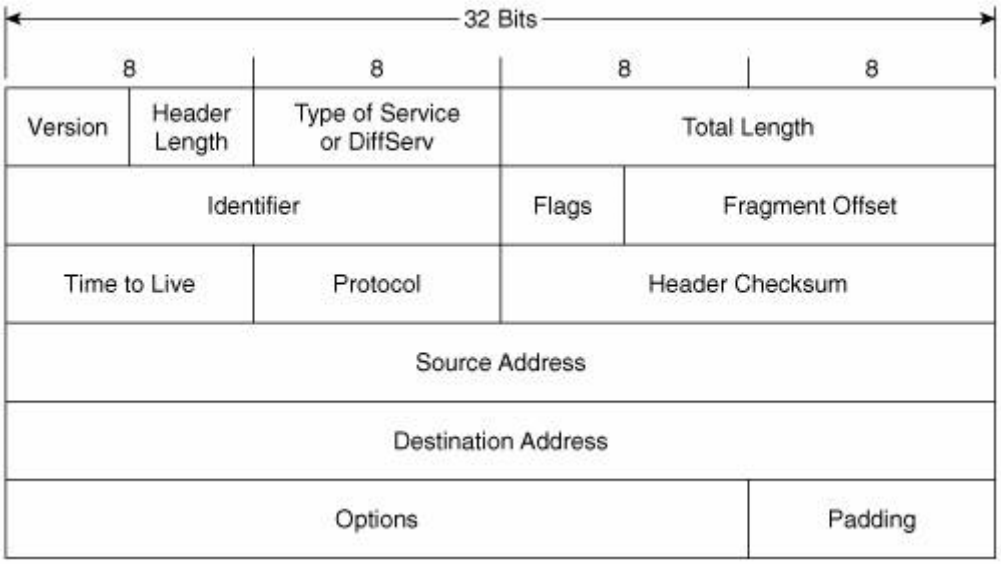
BAB II
TINJAUAN PUSTAKA

2.1 IPv4 *(Internet Protocol version 4)*

IP versi 4 adalah sebuah jenis pengalamatan jaringan yang digunakan di dalam protokol jaringan TCP/IP yang menggunakan protokol IP versi 4. Panjang totalnya adalah 32-bit, dan secara teoritis dapat mengalamati hingga 4 miliar host komputer atau lebih tepatnya 4.294.967.296 host di seluruh dunia, jumlah host tersebut didapatkan dari 256 (didapatkan dari 8 bit) dipangkat 4(karena terdapat 4 oktet) sehingga nilai maksimal dari alamt IP versi 4 tersebut adalah 255.255.255.255 dimana nilai dihitung dari nol sehingga nilai nilai host yang dapat ditampung adalah 256x256x256x256=4.294.967.296 host, bila host yang ada di seluruh dunia melebihi kuota tersebut maka dibuatlah IP versi 6 atau IPv6.

2.1.1 Format Header IPv4



Gambar 2.1 Format Header IPv4

1. Version → Mengindikasikan versi IP yang digunakan. Field ini berukuran 4-bit.
2. IP Header length → Menunjukkan ukuran header yang digunakan dalam satuan per 4 bytes.
3. Type of Service → Field ini menunjukkan layanan yang hendak dipakai oleh paket yang bersangkutan.
4. Total Length → Menunjukkan ukuran paket yang terdiri dari header dan data.

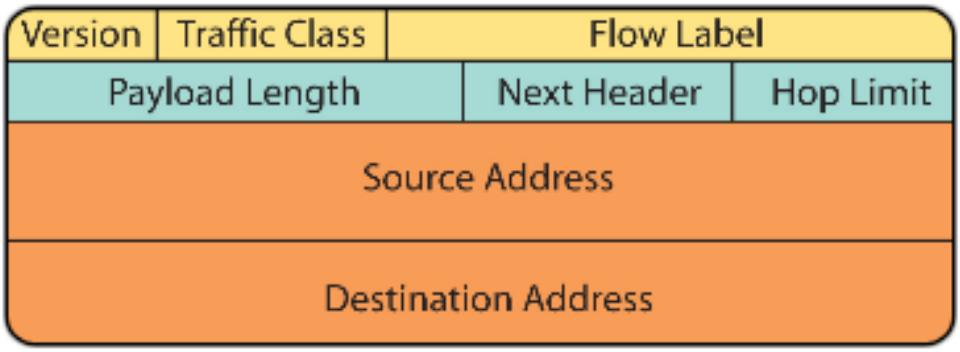
5. Identification → Menunjukkan identitas suatu fragmen yang digunakan dalam penyatuan kembali (reassembly) menjadi paket utuh.
6. Flags → Menunjukkan tanda- tanda tertentu dalam proses fragmentasi.
7. Fragment Offset →Menunjukkan posisi setiap fragmen.
8. Time to Live → Menunjukkan jumlah node maksimal yang dapat dilalui oleh setiap paket yang dikirim.
9. Protocol → Menunjukkan protocol di lapisan yang lebih tinggi.
10. Header Checksum → Menunjukkan nilai yang digunakan dalam pengecekan kesalahan terhadap header sebelum dengan sesudah pengiriman.
11. Source Address →Menunjukkan alamat pengirim paket.
12. Destination Address →Menunjukkan alamat penerima paket.
13. Option →Menunjukkan informasi yang memungkinkan suatu paket meminta layanan tambahan
14. Padding →Bit bit "0" tambahan yang ditambahkan ke dapam field ini untuk memastikan header IPv4 tetap berukuran multiple 32-bit.

2.2. IPv6 *(Internet Protocol version 6)*

IPv6 dikembangkan oleh IETF untuk dapat memenuhi kebutuhan IP yang diperlukan, selain itu IPv6 juga dikembangkan untuk mengatasi atau menyempurnakan kekurangan-kekurangan dari teknologi pendahulunya, yaitu IPv4. Kelebihan utama dari IPv6 adalah pengalamatannya yang luas, yaitu 128- bit. Dengan demikian ada 2128 atau sekitar 3,4 x 1038 alamat IPv6 yang berlimpah tersedia, sehingga dapat memenuhi kebutuhan IP saat ini maupun di masa mendatang. IPv6 ini dapat mengatasi masalah pada NAT (Network Address Translation) yang mengurangi atau menghalangi penggunaan dari aplikasi realtime yang membutuhkan hubungan dua arah. Pada IPv6 mendukung hirarki pengalamatan dan jumlah pengalamatan node yang lebih banyak, sehingga konfigurasi alamat lebih sederhana. Kelebihan dari IPv6 yang lain ialah kapabilitas untuk QOS *(Quality Of Service)*, autentifikasi, dan privasi. Untuk QOS dimungkinkan untuk pemberian label pada paket-paket pada aliran trafik tertentu yang membutuhkan penanganan khusus, untuk autentifikasi dan privasi IPv6 mendukung autentifikasi, integritas data, dan kerahasiaan data.

2.2.1. Format Header IPv6

Pada IPv6 digunakan header paket yang sederhana, dan dengan header yang sederhana paket dapat diproses secara lebih efisien. Header pada IPv6 merupakan penyederhanaan dari header IPv4 dengan menghilangkan bagian yang tidak dipergunakan atau jarang digunakan dan menambahkan bagian yang menyediakan dukungan yang lebih baik untuk keperluan mendatang. Pada Gambar 2.1 dian format header pada IPv6.



Gambar 2.2 Header pada IPv6

Berikut penjelasan dari format header pada IPv6 :

- *Version* → Versi IP, enam untuk IPv6 (4-bit).
- *Traffic Class* → Klasifikasi trafik, field ini menentukan prioritas trafik atau paket dan digunakan untuk QOS (quality of service) (8-bit).
- *Flow Label* → Label aliran dari trafik (20-bit).
- *Payload Length* → Field ini merupakan panjang dari paket data (16-bit).
- *Next Header* → Identifikasi tipe header setelah header IPv6 (8-bit).
- *Hop Limit* → Nilai akan dikurangi satu, jika melewati sebuah router (node) .
- *Source/ Destination Address* → Alamat dari sumber dan tujuan.
- *Extension Header* → Sebagai informasi tambahan, yang ditempatkan diantara header IPv6 dengan header yang lebih tinggi diatasnya.

2.2.2. Pengalamatan pada IPv6

Pada IPv6 alamat ditulis dalam format hexadesimal dengan pemisah berupa titik dua diantara masing-masing 16-bit. Format penulisan alamat IPv6 adalah x : x : x : x : x : x : x : x dimana x adalah empat digit bilangan heksadesimal. Ada beberapa aturan dalam penulisan atau meringkas alamat IPv6, sebagai contoh untuk alamat

2033:0000:140E:0000:0000:09D0:683A:140A, ada beberapa yang dapat disederhanakan, dengan cara sebagai berikut :

- Angka nol (0) di awal adalah optional. Sebagai contoh pada 09DO disederhanakan menjadi 9D0, dan 0000 dapat disederhanakan menjadi 0, sehingga 2033:0000:140E:0000:0000:09D0:683A:140A dapat ditulis menjadi 2033:0:140E:0000:0000:9D0:683A:140A.
- Angka nol yang berurutan dapat disederhanakan dengan dua tanda titik dua "::". Namun penyederhanaan ini hanya dapat digunakan satu kali dalam sebuah alamat. Sebagai contoh 2033:0:140E:0000:0000:9D0:683A:140A, dapat ditulis menjadi 2033:0:140E::9D0:683A:140A.

Arsitektur pengalamatan pada IPv6 dibagi tiga, yaitu :

- *Unicast Address*

Unicast address adalah alamat yang menunjuk pada sebuah alamat antarmuka atau *host*. Pada alamat *unicast* dibagi menjadi menjadi 3, yaitu : alamat *link local* alamat yang digunakan dalam satu link jaringan, alamat *site local* setara dengan alamat *private* pada IPv4, dan alamat global, yaitu alamat *public* yang digunakan oleh ISP (*Internet Service Provider*).

- *Multicast Address*

Multicast address adalah alamat yang menunjukan beberapa *interface* (biasanya untuk node yang bebeda). Paket yang dikirimkan ke alamat ini akan dikirimkan ke semua *interface* yang dian oleh alamat ini. Alamat *multicast* didesain untuk menggantikan alamat *broadcast* pada IPv4.

- *Anycast Address*

Anycast address adalah alamat yang menunjukkan beberapa *interface* (biasanya untuk node yang bebeda). Paket yang dikirimkan ke alamat ini akan dikirimkan ke salah satu alamat *interface* yang paling dekat dengan *router*.

2.3. *Routing Protocol*

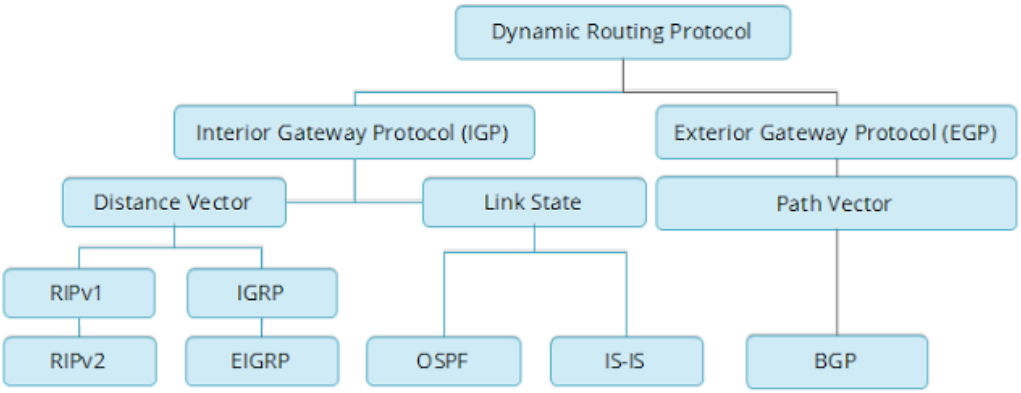
Routing adalah suatu protokol yang digunakan untuk mendapatkan rute atau petunjuk dari satu jaringan ke jaringan yang lain, *routing* merupakan proses dimana suatu router akan memilih jalur atau rute untuk mengirimkan atau meneruskan suatu paket ke jaringan yang

dituju. Router menggunakan IP address tujuan untuk mengirimkan paket, dan agar router mengetahui rute mana yang harus digunakan untuk meneruskan paket ke alamat tujuan, router harus belajar atau bertukar informasi sesama router yang saling terhubung untuk mengetahui jalur atau rute yang terbaik.

Routing protocol digunakan untuk memfasilitasi pertukaran informasi *routing* antar router. Dengan *routing protocol*, router dapat berbagi informasi *routing table,* yaitu informasi mengenai jaringan lain yang saling terhubung. Ada beberapa *routing protocol* yang mendukung IPv6, yaitu RIPng, OSPFv3 EIGRP for IPv6 (Cisco *properiarity)*, IS-IS for IPv6, BGP IPv6, dan lainnya. Masingmasing dibuat berdasarkan *routing protocol* sebelumnya yang mendukung IPv4 namun disesuaikan dengan lingkup IPv6 dan memiliki beberapa kelebihan dan pembaharuan serta cara konfigurasi yang berbeda pada router.

2.3.1. Klasifikasi protokol *dynamic routing*

Pada protokol *routing* kelas *Interior Gateway Protocols* (IGPs) *dynamic routing* diklasifikasi menjadi dua, yaitu *distance vector routing* dan *link-state routing*. Untuk klasifikasi *dynamic routing protocol* secara keseluruhan terlihat seperti pada Gambar 2.2. Pembagian pada *dynamic routing protocol* dibedakan berdasarkan karakteristik dan cara kerjanya masing-masing.



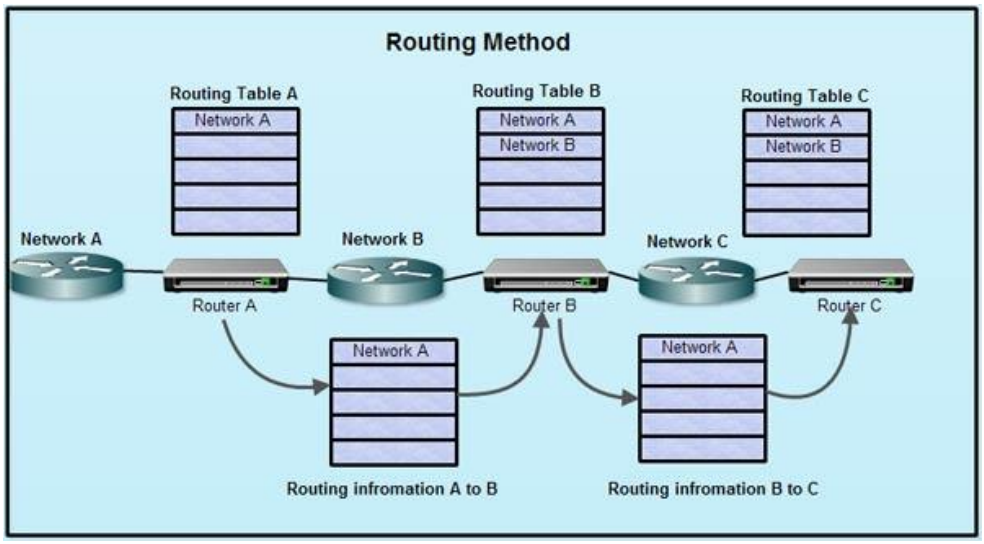
Gambar 2.3 Klasifikasi *dynamic routing protocol*

2.3.1.1. *Distance vector routing*

Router yang menggunakan jarak dan arah sebagai acuan *routing* dinamakan *distance vector routing*. Pada *distance-vector routing protocol* digunakan algoritma *Bellman-Ford* dalam kalkulasi untuk pemilihan jalur. Informasi atau *update table* pada *distance-vector routing protocol* dilakukan secara berkala oleh router, berbeda dengan link-state yang melakukan *update*

table setiap ada perubahan pada topologi jaringan, sehingga pada *distance-vector routing protocol* membutuhkan proses komputasi yang lebih sederhana. Contoh *routing protocol* yang menggunakan *distance-vector routing protocol* adalah RIPv1, RIPv2, dan IGRP.

Seperti namanya, maka pada *distance-vector routing protocol* menggunakan jarak dan arah untuk melakukan *routing*. Jarak yang dimaksud adalah *hop count* atau jumlah router yang dilalui, dan untuk arah yang dimaksud adalah alamat *next hop* atau *interface* keluar yang digunakan oleh router. Pembaharuan atau Update dilakukan secara berkala pada distance-vector routing protocol dimana update routing table dikirimkan ke semua router yang bersebelahan secara langsung yang dikonfigurasi dengan distance-vector routing protocol yang sama.



Gambar 2.4 *Distance-vector routing protocol*

2.3.1.2. *Link-state routing protocol*

Link-state routing protocol dibangun dengan algoritma Edsger Dijkstra's atau kadang disebut algoritma *shortest path first* (SPF). Algoritma ini menjumlahkan total *cost* yang dibutuhkan pada masing-masing jalur dari alamat asal ke alamat tujuan. *Link-state routing protocol* membangun suatu topologi jaringan, dimana masing-masing router yang terhubung menggunakan gambaran topologi tersebut untuk menentukan jalur atau rute untuk menjangkau jaringan yang ingin dicapai. Router dengan *link-state* akan mengirimkan kondisi dari *link*nya ke router-router lain yang berada dalam *routing domain* yang sama. Informasi atau kondisi *link* yang disebarkan adalah kondisi link pada router yang terhubung langsung suatu jaringan dan kondisi *link* pada router yang saling terhubung. Router dengan

link-state routing protocols menggunakan *Hello protocol* untuk mengetahui *link-link* yang terhubung dengan router tetangga atau router yang terhubung langsung.

Pada *link-state routing protocol* ada beberapa kelebihan bila dibandingkan dengan *distance-vector routing protocol*, seperti membangun peta topologi jaringan, sehingga masing-masing router dapat menentukan sendiri jalur yang pendek untuk mencapai jaringan yang lain. Konvergensi jaringan terjadi dengan cepat, karena ketika router menerima paket LSP langsung disebar ke router tetangganya yang lain dalam jaringan. *Update* atau pembaharuan informasi dilakukan saat terjadi perubahan pada *link* secara langsung. Disain secara hirarki, dimana *link-state routing protocols* menggunakan konsep area, dan area-area disusun secara hirarki, sehingga *routing* lebih baik. Namun *link-state routing protocol* juga memiliki kekurangan, dimana *routing protocol* ini membutuhkan kinerja *CPU*, *memory*, dan *bandwith* yang lebih besar.

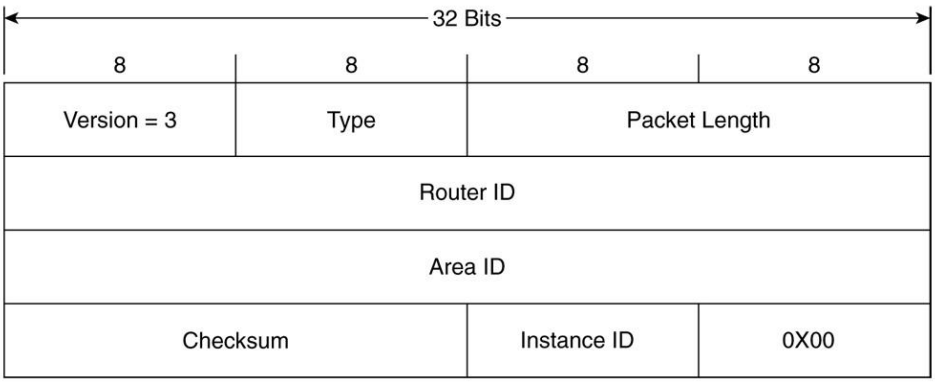
2.3.2. OSPF

Routing Protocol Open Shortest Path First (OSPF) adalah sebuah *routing protocol* standard terbuka yang telah diimplementasikan oleh sejumlah besar vendor jaringan. Alasan untuk mengkonfigurasi OSPF dalam sebuah topologi adalah untuk mengurangi overhead (waktu pemrosesan) routing, mempercepat convergance,serta membatasi ketidakstabilan network disebuah area dalam suatu network. OSPF Message Encapsulation terjadi pada lapisan data-link dengan nomor protocol 89. Data field ini dapat berisi salah satu dari lima tipe paket OSPF. Pada IP packet header, alamat tujuannya mempunyai dua alamat multicast yaitu 224.0.0.5 dan 224.0.0.6 namun yang diset cukup salah satu dari alamat tersebut. Semua paket OSPF mempunyai 24 byte yang berisikan informasi yang diperlukan. *Packet* header ini terdiri dari berbagai bidang seperti jenis-jenis *packet* OSPF, *router* ID serta alamat IP dari router yang mengir imkan paket.

2.3.3. OSPFv3

Open Shortest Path First (OSPF) adalah *routing protocol* kelas *link-state* yang dikembangkan untuk memperbaiki kinerja dari *routing protocol* RIP. OSPF adalah *routing protocol* yang menggunakan konsep *area*. Kelebihan dari OSPF dibandingkan dengan RIP adalah kecepatan dalam melakukan konvergensi dan lebih luasnya jaringan yang bisa dijangkau. Pada dasarnya OSPFv3 menggunakan jenis paket yang sama pada OSPFv2. Perbedaan yang paling jelas ialah OSPFv3 mendukung pengalamatan 128-bit. OSPFv2 menggunakan alamat

224.0.0.5 dan 224.0.0.6, OSPFv3 menggunakan alamat multicast IPv6 yaitu FF02::5 dan alamat FF02::6 untuk router DR (*designated routers*) dan BDR (*Backup DRs*). OSPFv3 menggunakan alamat *link-localnya* untuk malakukan *advertisements* bukan alamat globalnya. Paket header OSPFv3 adalah sebesar 16-byte, berbeda dengan OSPFv2 sebesar 24-byte. Paket header OSPFv3 terlihat seperti pada Gambar 2.4.



Gambar 2.5 Paket header OSPFv3

Dari Gambar 2.4 terlihat pada paket header OSPFv3 tidak ada autentikasi. Pada IPv6 kemampuan dalam autentikasi dan enkripsi menggunakan header extension. Pada OSPF terdapat beberapa paket LSP (Link-State Packets), masing-masing paket dibutuhkan dalam proses routing pada OSPF. Berikut paketpaket LSP pada OSPF:

1. Hello – Paket Hello digunakan untuk memulai dan menjaga keterhubungan informasi dengan router OSPF yang lain.
2. DBD (Paket Database Description) – DBD untuk memeriksa dan melakukan sinkronisasi database antar router.
3. LSR (Link-State Request) – LSR digunakan untuk menarik informasi dari router lain.
4. LSU (Link-State Update) – Paket ini digunakan untuk menjawab LSR
5. LSAck (Link-State Acknowledgment) – LSAck digunakan untuk mengkonfirmasi paket LSU yang diterima oleh router.

Masing-masing router OSPF menjaga database LSA yang diterima dari router lain. Ketika LSA dari semua router telah diterima maka router akan membangun sebuah local link-state database. OSPF menggunakan algoritma Dijkstra's shortest path first (SPF) untuk membangun sebuah SPF tree. SPF tree ini yang kemudian digunakan untuk membangun sebuah routing table dengan jalur terbaik guna mencapai jaringan yang lain.

Berikut merupakan proses terjadinya konvergensi pada *link-state routing protocols* :

1. Masing-masing router mempelajari koneksinya yang terhubung ke jaringan secara langsung.

2. Tiap router bertanggung jawab untuk "*Hello*" ke router tetangga yang terhubung langsung.
3. Router membangun *Link-State Packet* (LSP) yang berisi mengenai informasi *link* yang terhubung langsung.
4. Masing-masing router akan mengirimkan LSP kesemua tetangganya, yang kemudian disimpan pada *database.*
5. Tiap router menggunakan *database*nya untuk membangun sebuah peta topologi lengkap dari jaringan dan gambaran jalur atau rute yang dapat digunakan untuk mencapai jaringan tujuan yang ingin dicapai.

Administrative distance (AD) digunakan untuk mengukur realibilitas informasi *routing* yang diterima oleh sebuah router dari router tetangganya. Nilai AD berkisar pada bilangan bulat antara 0 sampai 255, dimana 0 menunjukkan kemampuan penerusan data yang tertinggi dan 255 menunjukan tidak ada data yang akan diteruskan melewati sebuah rute. Secara *default* OSPF memiliki AD bernilai 110. *Metric* yang digunakan pada *routing protocol* OSPF dinamakan *cost,* semakin kecil nilai *cost,* maka akan dipilih menjadi *interface* untuk mengirimkan data. Pada RFC 2328 tidak dijelaskan mengenai nilai acuan untuk *cost,*

> *RFC 2328 : "A cost is associated with the output side of each router interface. This cost is configurable by the system administrator. The lower the cost, the more likely the interface is to be used to forward data traffic.*

Namun pada Cisco IOS akumulasi *bandwith* dari *interface* yang digunakan router untuk mencapai tujuan dijadikan sebagai acuan nilai *cost*. Pada setiap router nilai cost dari *interfacenya* dihitung dari 10 pangkat 8 dibagi dengan nilai *bandwidth*nya (bps).

Tabel 2.1 Nilai cost pada OSPF Cisco

Interface Type	Cost
Fast Ethernet and faster	100000000 bps = 1
Ethernet	10000000 bps = 10
E1	2048000 bps = 48
T1	1544000 bps = 64
128 kbps	128000 bps = 781
64 kbps	64000 bps = 1562
56 kbps	56000 bps = 1785

OSPF memiliki 3 tabel di dalam router :

1. ***Routing table*** biasa juga disebut sebagai *Forwarding database*. Database ini berisi the *lowest cost* untuk mencapai *router-router/network-network* lainnya. Setiap *router* mempunyai *Routing table* yang berbeda-beda.
2. ***Adjecency database,*** *Database* ini berisi semua *router* tetangganya. Setiap *router* mempunyai *Adjecency database* yang berbeda-beda.
3. ***Topological database,*** *Database* ini berisi seluruh informasi tentang *router* yang berada dalam satu networknya/areanya.

2.4. *Router*

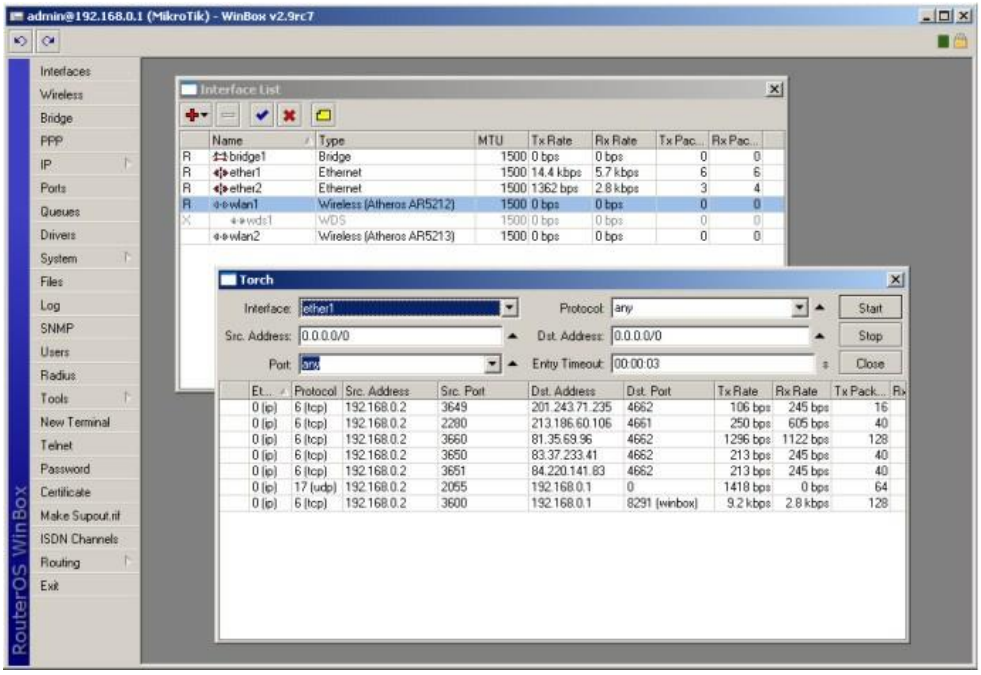
Router adalah sebuah alat jaringan komputer yang mengirimkan paket data melalui sebuah jaringan atau internet menuju tujuannya, melalui sebuah proses yang dikenal sebagai routing.

Router berfungsi sebagai penghubung antar dua atau lebih jaringan untuk meneruskan data dari satu jaringan ke jaringan lainnya. *Router* disebut sebagai peralatan jaringan yang meneruskan suatu paket data/informasi dan memilih rute terbaik untuk ditempuh untuk menyimpulkan data/informasi tersebut.

Gambar 2.6 Perangkat *Router*

2.5 *Winbox*

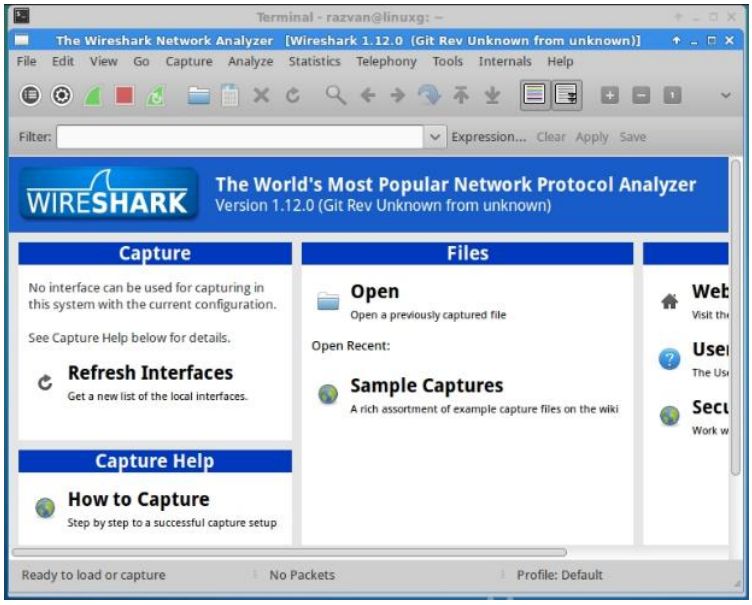
Winbox adalah sebuah software atau utility yang di gunakan untuk meremote sebuah server mikrotik kedalam mode GUI (Graphical User Interface) melalui *operating system* windows. Kebanyakan teknisi banyak mengkonfigurasi mikrotik os atau mikrotik *router board* menggunakan *winbox* di banding dengan yang mengkonfigurasi langsung lewat mode CLI *(Command Line Interface)*. Hal ini karena menggunakan *winbox* dirasa lebih mudah dan simple dibanding melalui browser. Dan hasilnya pun juga lebih cepat.



Gambar 2.7 Tampilan pada *Winbox*

2.6 *Wireshark*

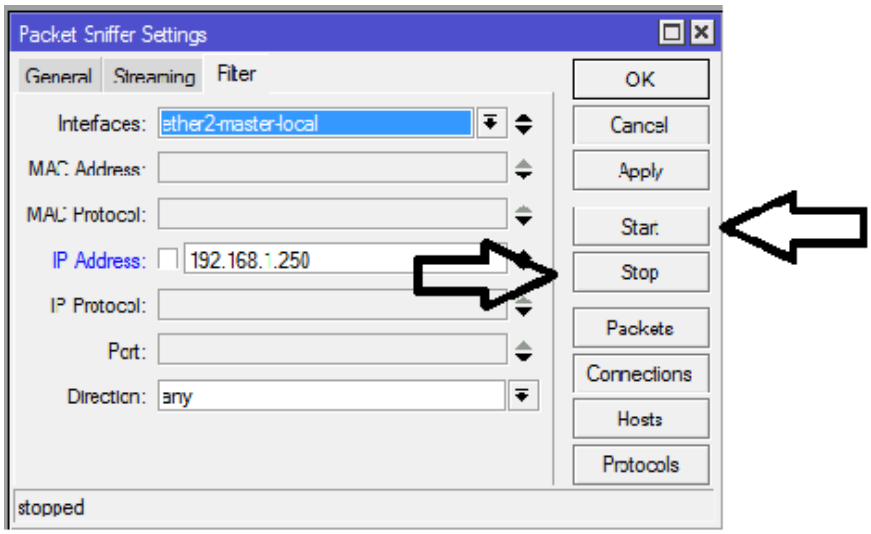
Wireshark adalah suatu perangkat lunak yang digunakan untuk analisa paket pada jaringan, *Wireshark* yang digunakan pada pengujian adalah versi 2.0.0. Dengan perangkat lunak ini dapat dilakukan analisis, *troubleshooting*, penelitian protokol, dan lainnya. *Wireshark* mampu menangkap paket-paket data atau informasi yang melewati jaringan. Berbagai format protokol dapat ditangkap dan dianalisa. Perangkat lunak ini bekerja dengan melakukan *capture* atau manangkap paket data melalui interface pada PC yang terhubung pada jaringan.



Gambar 2.8 Tampilan pada *Wireshark*

2.7 *Mikrotik Packet Sniffer*

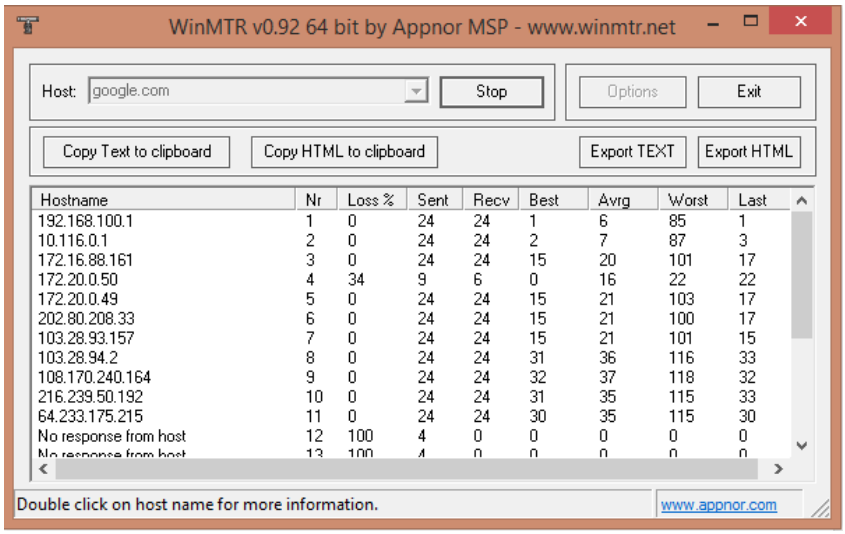
Untuk melakukan capture paket pada interface antar router digunakan perangkat lunak Packet Sniffer yang tersedia pada router Mikrotik. Tool ini disediakan dalam mikrotik untuk menangkap paket-paket data yang melalui interface router pada jaringan (paket yang masuk dan keluar melalui router) . Tool ini berguna untuk analisis pada *traffic* jaringan. Untuk menggunakan tool ini cukup dengan mengetikan perintah "tool sniffer > start" pada terminal Mikrotik di masing-masing router dan dipilih *interface* apa yang ingin di c*apture*.



Gambar 2.9 Tampilan pada *Mikrotik Packet Sniffer*

2.8. WinMTR

WinMTR adalah sebuah program under windows yang populer dibuat oleh Matt. WiMTR merupakan kombinasi dari dua perintah di windows yaitu: ping dan tracert .



Gambar 2.10 Tampilan pada WinMTR