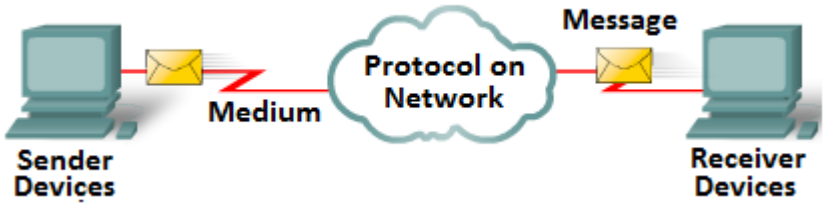
BAB 2
LANDASAN TEORI

2.1 Komunikasi

Komunikasi merupakan salah satu tujuan dibentuknya suatu jaringan. Komunikasi dalam jaringan komputer merupakan proses perpindahan data atau informasi yang terjadi diantara setiap *host* komputer yang terhubung. Secara sederhana, komunikasi pada jaringan diilustrasikan sebagai berikut:



Gambar 2.1 Elemen-Elemen Komunikasi [6]

Elemen-elemen untuk mendukung terjadinya suatu komunikasi [6], yaitu:

1. Media Pengirim (*Sender Devices*)
   Merupakan piranti yang melakukan pengiriman data.
2. Media Penerima (*Receiver Devices*)
   Merupakan piranti yang menerima data.
3. Data (*Message*)
   Merupakan informasi yang dipindahkan.
4. Media Transmisi (*Medium*)
   Merupakan saluran yang digunakan pada proses pengiriman data, media transmisi dapat berupa media kabel maupun nirkabel.
5. Protokol (*Protocol*)
   Merupakan aturan-aturan yang berfungsi untuk menyelaraskan hubungan pada saat proses pengirimian dan penerimaan data.

2.2 Jaringan Komputer

Sejarah jaringan komputer berawal dari *time-sharing networks*, yaitu ‘rangkaian’ terminal yang terhubung dengan komputer sentral yang disebut *mainframe*. Contoh *time-sharing networks* adalah *IBM’s System Network Architecture (SNA)* dan *Digital Network Architecture* [1].

Tujuan dari jaringan komputer adalah untuk melakukan komunikasi data, *sharing* data maupun pemakaian *resource* bersama seperti printer dan media penyimpanan sekunder. Komunikasi data sendiri memiliki tujuan yang lebih khusus [7], yaitu:

1. Memungkinkan pengiriman data dalam jumlah besar, efisien, ekonomis, dan tanpa kesalahan dari suatu tempat ke tempat yang lain.
2. Memungkinkan penggunaan sistem komputer dan peralatan pendukung dari jarak jauh (*remote*).
3. Memungkinkan penggunaan komputer secara terpusat maupun secara tersebar sehingga mendukung manajemen dalam hal kontrol, baik desentralisasi ataupun sentralisasi.
4. Mempermudah kemungkinan pengelolaan dan pengaturan data yang ada dalam berbagai macam sistem komputer.
5. Mengurangi waktu untuk pengelolaan data.
6. Mendapatkan data langsung dari sumbernya.
7. Mempercepat penyebarluasan informasi.
8. Komunikasi data berkaitan dengan pertukaran data diantara dua perangkat yang terhubung secara langsung yang memungkinkan adanya pertukaran data antar kedua pihak.

2.3 Perangkat Jaringan

Perangkat jaringan merupakan alat atau piranti yang digunakan untuk membangun suatu sistem jaringan. Masing-masing perangkat jaringan memiliki fungsi dan tujuan tersendiri didalam suatu sistem jaringan. Pemilihan perangkat-perangkat jaringan yang diperlukan dapat disesuaikan dengan kebutuhan sistem jaringan yang akan dibangun.

2.3.1 *Router*

Router sering digunakan untuk menghubungkan beberapa *network*. Baik *network* yang sama maupun yang berbeda dari segi teknologinya. Seperti menghubungkan *network* yang menggunakan topologi *Bus*, *Star*, dan *Ring*. *Router* juga digunakan untuk membagi *network* besar menjadi beberapa buah *subnetwork* (*network-network* kecil). Setiap *subnetwork* seolah-olah "terisolir" dari *network* lain. Hal ini dapat membagi-bagi *traffic* yang akan berdampak positif pada performa *network* [1].

Sebuah *router* memiliki kemampuan *routing*. Artinya *router* secara cerdas dapat mengetahui kemana rute perjalanan informasi (yang disebut *packet*) akan dilewatkan. Apakah ditujukan untuk *host* lain dalam satu *network* ataukah berbeda network. Jika paket-paket ditujukan untuk *host* pada *network* yang sama maka *router* akan menghalangi paket-paket keluar, sehingga paket-paket tersebut tidak "membanjiri" *network* yang lain [1].

2.3.2 *Bridge*

Bridge atau kadangkala disebut *transparent bridge* merupakan perangkat *network* yang digunakan untuk menghubungkan dua buah LAN atau membagi sebuah LAN menjadi dua buah segmen. Tujuannya adalah untuk mengurangi *traffic* sedemikian rupa sehingga dapat meningkatkan performa *network* [1].

Bridge dan *router* secara sepintas terlihat sama dan dapat menghubungan dua buah LAN. Namun sesungguhnya cara kerja dan fungsi utama kedua perangkat tersebut berbeda [1].

Tabel 2.1 Perbandingan *Router* dan *Bridge* [1]

Router	**Bridge**
Mendukung berbagai *network protocol address*, seperti IP, IPX, *Apple Talk.*	Tidak mendukung *network protocol address*. Hanya mengenali MAC *address*.

Dapat menghubungkan beberapa subnet yang menggunakan teknologi yang berbeda-beda.	Menghubungkan dua buah segmen. Semua segmen dipandang sebagai sebuah subnet.
Mampu memblok *traffic* antar subnet.	Tidak dapat memblok *traffic* dari subnet lain.
Cocok digunakan pada sembarang protokol *network*.	Cocok digunakan pada protokol *non-routable* seperti NetBIOS dan DECnet.
Instalasi dan konfigurasi memerlukan keahlian khusus.	Instalasi relatif mudah, pasang dan nyalakan.
Cocok digunakan pada Internet dan Intranet.	Cocok digunakan pada LAN atau Intranet.

2.3.3 *Switch*

Dalam sebuah jaringan, *switch* berfungsi sebagai sentral atau konsentrator pada sebuah *network*. Cara kerja *switch* mirip dengan *bridge*, sehingga *switch* disebut sebagai *multiport bridge*. Ada dua arsitektur dasar yang digunakan pada switch, yaitu *cut-through* atau *fast forward* dan *store and forward*.

Cut-through merupakan jenis *switch* 'tercepat' diantara jenis lainnya. Switch jenis ini hanya mengecek alamat tujuan (yang ada pada *header frame*). Selanjutnya *frame* akan diteruskan ke *host* tujuan. Kondisi ini dapat mengurangi 'waktu tunggu' atau *latency*. Karena hanya mengecek alamat tujuan, kelemahan switch jenis ini yaitu tidak dapat mengecek *frame-frame* yang *error*. *Frame* yang *error* akan tetap diteruskan ke *host* tujuan [1].

Store and forward switch merupakan jenis *switch* yang akan menyimpan semua *frame* untuk sementara waktu sebelum diteruskan ke *host* tujuan. Seluruh *frame* akan dicek melalui mekanisme CRC (*Cyclic Redundancy Check*). Jika ditemukan *error* maka *frame* akan 'dibuang' dan tidak diteruskan ke *host* tujuan. *Switch* jenis ini paling 'terpercaya' diantara jenis lainnya. Kelemahan *switch* jenis

ini adalah meningkatnya *latency* akibat adanya proses pengecekan seluruh *frame* yang melalui *switch* [1].

2.3.4 *Hub*

Hub mirip dengan *switch* yaitu sebagai konsentrator. Namun, *hub* tidak 'secerdas' *switch*. Jika informasi dikirim ke *host* target melalui hub maka informasi akan mengalir ke semua *host*. Kondisi semacam ini dapat menyebabkan beban *traffic* yang tinggi. Oleh sebab itu, sebuah *hub* biasanya hanya digunakan pada *network* berskala kecil [1].

2.3.5 *Repeater*

Repeater termasuk satu dari perangkat keras jaringan komputer yang dipasang di titik-titik tertentu dalam jaringan untuk memperbaharui sinyal yang ditransmisikan agar mencapai kembali kekuatan dan bentuknya semula guna memperpanjang jarak tempuh. *Repeater* berfungsi untuk menguatkan sinyal.

2.3.6 *Network Interface Card*

Kartu Jaringan (NIC) merupakan salah satu dari perangkat keras yang menyediakan media untuk menghubungkan antar komputer. Kebanyakan kartu jaringan adalah kartu internal, yaitu kartu jaringan yang di pasang pada slot ekspansi di dalam komputer. Kartu jaringan yang banyak terpakai saat ini adalah: kartu jaringan *Ethernet*, *LocalTalk* konektor, dan kartu jaringan *Token Ring*.

2.4 Topologi Jaringan

Topologi jaringan merupakan suatu aturan atau cara untuk menghubungkan komputer yang satu dengan komputer yang lainnya sehingga membentuk suatu jaringan. Topologi jaringan dapat didefinisikan juga sebagai gambaran secara fisik dari pola hubungan antar komponen jaringan. [8]

Ada beberapa jenis topologi jaringan yang dikenal secara luas yang memiliki kelebihan dan kekurangan tersendiri. Topologi jaringan juga sangat berpengaruh terhadap performasi maupun kompleksitas jaringan yang dibangun,

oleh karena itu dalam pemilihan topologi jaringan harus disesuaikan dengan kebutuhan terhadap kesesuaian infrastruktur jaringan yang akan dibangun.

2.4.1 Faktor-Faktor Yang Mempengaruhi Pemilihan Topologi Jaringan

Untuk membangun sebuah topologi jaringan, perlu diperhatikan beberapa faktor yang akan mempengaruhi proses yang terjadi dalam topologi jaringan yang akan dibangun, faktor-faktor tersebut yaitu:

1. Biaya.

   Biaya merupakan faktor mendasar untuk memutuskan dengan teknologi seperti apa topologi jaringan akan dibangun.
2. Kecepatan

   Kecepatan pada jaringan sangat erat kaitannya dengan efektifitas dan efisiensi pengalokasian sumberdaya *bandwidth* yang tersedia terhadap kesesuaian kecepatan yang dibutuhkan dalam sistem.
3. Lingkungan

   Pembangunan infrastruktur jaringan sangat dipengaruhi lokasi dimana sistem tersebut dibangun sehingga perlu diperhatikan juga dalam pemilihan perangkat-perangkat jaringan yang digunakan
4. Ukuran

   Ukuran dilihat dari sejauh mana inftastruktur jaringan akan dibangun apakan berskala *enterprise* dengan memiliki banyak *server* ataukah tidak.
5. Konektivitas.

   Keandalan dari suatu infrastruktur jaringan juga perlu untuk diperhatikan guna mempermudah akses terhadap jaringan yang ada.

2.4.2 Konsep Topologi Jaringan

Dalam membangun sebuah topologi jaringan tidak selalu perangkat-perangkat yang terhubung mewakili cara akses dari setiap perangkat jaringan yang terhubung itu. Cara akses dan keterhubungan antar komponen jaringan dapat

dipandang sebagai dua jenis topologi yang berbeda yang dilihat secara fisik dan *logic*.

Topologi fisik jaringan adalah cara yang digunakan untuk menghubungkan *workstation-workstation* di dalam LAN [9] atau dapat disebut sebagai suatu desain posisi dan letak *node* yang dilihat secara nyata untuk menentukan keterhubungan antar komponen jaringan melalui kabel atau media lainnya.

Sementara topologi logic jaringan adalah topologi yang dilihat dari metode accessnya [9] atau dapat disebut sebagai desain jaringan komputer untuk mentransmisikan data antar komputer yang dilihat dari cara kerjanya.

Sehingga dapat disimpulkan bahwa topologi jaringan jika dilihat dari keterhubungan antar komponen jaringan dan cara transmisi data yang berlangsung antar komputer dapat dibedakan sebagai topologi fisik jaringan dan topologi *logic* jaringan.

2.5 *Open Systems Interconnection* (OSI)

Model OSI digunakan untuk menjelaskan cara kerja jaringan komputer secara logika. Walaupun OSI merupakan sebuah model yang diakui di dunia saat ini, namun tidak ada paksaan bagi pengembang *hardware* maupun *software* dan *user* untuk menggunakannya [1].

Model OSI terdiri atas 7 *layer* atau lapisan, berikut adalah tabel ketujuh *layer* tersebut berserta fungsinya:

Table 2.1 OSI Layer [1]

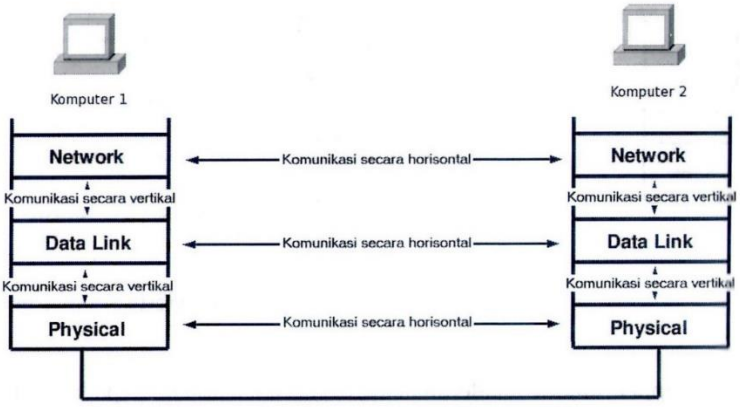
Layer	**Nama *Layer***	**Fungsi**
7	*Application*	Menyediakan *service* bagi berbagai aplikasi *network*
6	*Presentation*	Mengatur konversi dan translasi berbagai format data sepert kompresi data dan enkripsi data
5	*Session*	Mengatur sesi (*session*) yang meliputi *establishing* (memulai sesi), *maintaining* (mempertahankan sesi), dan

		terminating (mangakhiri sesi) antar entitas yang dimiliki oleh *presentation layer*
4	*Transport*	Menyediakan *end-to-end communication protocol*. Layer ini bertangggung jawab terhadap 'keselamatan data' dan 'segmentasi data', seperti: mengatur *flow control* (kendali aliran data), *error detection* (deteksi eror) *and correction* (koreksi), data *sequencing* (urutan data) dan *size of the packet* (ukuran paket)
3	*Network*	Mengatur rute yang dilalui oleh data. *Layer* ini menyediakan *logical addressing* (pengalamatan logika) dan *path determination* (penentuan rute tujuan)
2	*Data Link*	Menentukan pengalamatan fisik (*hardware address*), *error notification* (pendeteksi eror), *frame flow control* (kendali aliran *frame*) dan topologi *network*. Ada dua sublayer pada *data Link*, yaitu: *Logical Link Control* (LLC) dan *Media Access Control* (MAC) LLC mengatur komunikasi, seperti *error notification* dan *flow control*. Sedangkan MAC mengatur pengalamatan fisik yang digunakan dalam proses komunikasi antar-*adapter*.
1	*Physical*	*Layer* ini menentukan masalah kelistrikan/ gelombang/ medan dan berbagai prosedur/fungsi yang berkaitan dengan link fisik seperti besar tegangan/arus listrik, panjang maksimal media transmisi, pergantian fasa, jenis kabel dan konektor

2.5.1 Komunikasi Antar Layer

Komunikasi yang terjadi pada OSI *layer* dapat dilakukan secara vertikal dan horisontal. Secara vertikal, satu *layer* dapat berkomunikasi dengan *layer* yang berada tepat satu posisi diatasnya atau dibawahnya. Misalnya *layer Data Link* dapat berkomunikasi dengan *layer Network* atau *layer Physical*. Sementara

komunikasi secara horisontal dilakukan oleh *layer* dengan *layer* yang sama pada *host* lainnya [1].



Gambar 2.2 Komunikasi Antar Layer [1]

2.5.2 Enkapsulasi Data Antar Layer

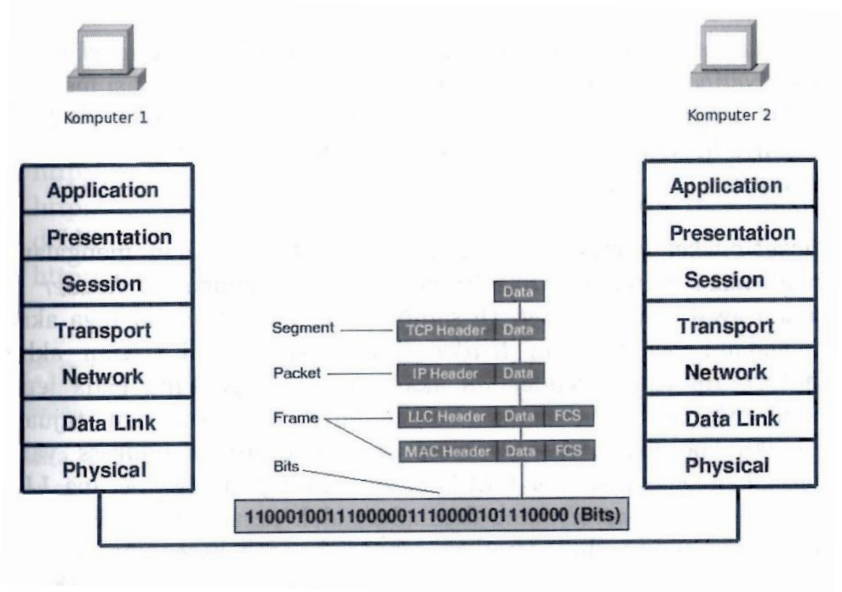
Data pada masing-masing *layer* mengalami perubahan bentuk atau transformasi yang disesuaikan dengan *layer* yang sedang dilewati. Ilustrasi yang menggambarkan perubahan bentuk yang terjadi pada data di setiap *layer* adalah sebagai berikut [1]:

1. Informasi berawal dari *layer Application*. Informasi kemudian melewati *layer Presentation* dan *layer Session*. Pada tahap ini biasanya belum dilakukan transformasi data. Informasi yang melalui ketiga layer ini disebut PDU (*Protocol Data Unit*) atau *data* saja.
2. Setelah sampai di *layer Transport*, data akan mengalami transformasi ke bentuk lain yang disebut *segment*.
3. *Segment* mengalir ke *layer Network* dan kemudian diubah menjadi *packet*.
4. *Packet* mengalir ke *layer Data Link* dan kemudian diubah menjadi *frame*.
5. Terakhir, *frame* mengalir ke *layer Physical* dan kemudian diubah menjadi *bits* atau bit-bit. Pada *layer* ini, bit-bit diubah menjadi besaran fisik, seperti arus listrik, gelombang elektromagnetik dan sebagainya.

Proses ‘pengubahan bentuk’ dari satu *layer* ke *layer* berikutnya dilakukan dengan menambahkan *header* khusus. Inilah yang disebut dengan *encapsulation* atau enkapsulasi. Proses enkapsulasi terjadi berulang-ulang hingga data diubah

menjadi bit-bit. Kemudian bit-bit tersebut dikirimkan ke *host* target melalui media jaringan.

Setelah informasi berupa bit-bit tersebut sampai di *host* target maka proses yang dilakukan oleh *host* target adalah *de-encapsulation* atau de-enkapsulasi yang dilakukan dengan cara melepas *header* satu persatu dimulai dari *layer* paling bawah hingga *layer* paling atas. Setelah proses *de-encapsulation* tersebut selesai maka informasi dapat disajikan oleh *host* target dengan aplikasi yang sesuai.



Gambar 2.3 Proses Enkapsulasi [1]

2.6 *Transmision Control Protocol / Internet Protocol* (TCP/IP)

TCP/IP adalah sekumpulan protokol yang terdapat di dalam jaringan komputer (*network*) yang digunakan untuk berkomunikasi atau bertukar data antar komputer. TCP/IP merupakan protokol standar pada jaringan internet yang menghubungkan banyak komputer yang berbeda jenis mesin maupun sistem operasi agar dapat berinteraksi satu sama lain [10].

Konsep TCP/IP berawal dari kebutuhan DoD (*Departement of Defense*) USA akan suatu komunikasi di antara berbagai variasi komputer yang telah ada. Komputer-komputer DoD ini seringkali harus menghubungkan antara satu organisasi peneliti dengan organisasi peneliti lainnya. Komputer tersebut harus

tetap berhubungan karena terkait dengan pertahanan negara dan sumber informasi harus tetap berjalan meskipun terjadi bencana alam besar, seperti ledakan nuklir. Oleh karenanya pada tahun 1969 dimulailah penelitian terhadap serangkaian protokol TCP/IP [10].

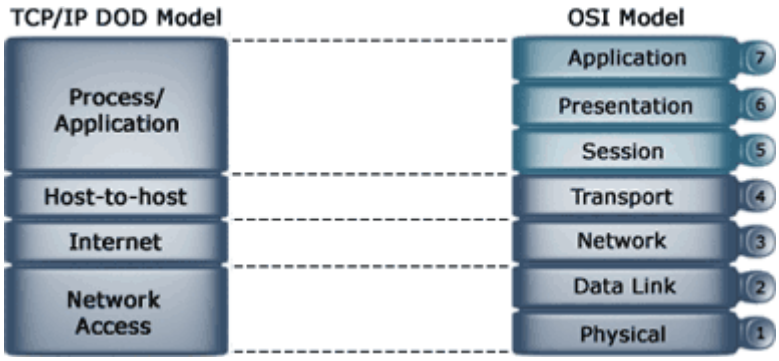
Adapun tujuan penelitian tersebut adalah sebagai berikut:

1. Terciptanya protokol-protokol umum, (DoD memerlukan suatu protokol yang dapat dipergunakan untuk semua jenis jaringan).
2. Meningkatkan efisiensi komunikasi data.
3. Dapat dipadukan dengan teknologi WAN (*Wide Area Network*) yang telah ada.
4. Mudah dikonfigurasi.

Tahun 1968 DoD ARPAnet (*Advanced Research Project Agency*) memulai penelitian yang kemudian menjadi cikal bakal *packet switching*. *Packet switching* inilah yang memungkinkan komunikasi antara lapisan *network*, dimana data dijalankan dan disalurkan melalui jaringan dalam bentuk unit-unit kecil yang disebut *packet*. Tiap-tiap paket ini membawa informasi alamat masing-masing yang ditangani dengan khusus oleh jaringan tersebut dan tidak tergantung dengan paket-paket lain. Jaringan yang dikembangkan ini adalah awal mula jaringan yang memiliki cakupan yang luas dan menjadi terkenal sebagi internet [10].

2.6.1 Arsitektur TPC/IP

Arsitektur rangkaian protokol TCP/IP didefinisikan dengan berbagai cara agar fungsi protokol-protokol TCP/IP tersebut dapat saling menyesuaikan. Protokol TCP/IP merupakan protokol standar yang terdapat pada Referensi Model DoD atau TCP/IP maupun Refernesi Model OSI. Jika model OSI memiliki 7 lapisan *layer* maka pada model TCP/IP memiliki 4 lapisan *layer* yang diperlihatkan sebagai berikut:



Gambar 2.4 TCP/IP Model dan OSI Model [11]

Hampir sama dengan OSI model, fungsi dari setiap layer pada TCP/IP model adalah sebagai berikut:

1. *Layer Application*

   Layer Application merupakan *layer* teratar dalam model TCP/IP *layer* ini bertanggung jawab untuk menyediakan akses kepada apliaksi terhadap layanan jaringan TCP/IP.
2. *Layer Host-to-Host* atau *layer Transport*

   Bertanggung jawab untuk komunikasi antar aplikasi dengan melakukan sesi koneksi yang bersifat *connection-oriented* (TCP) dan *connectionless* (UDP)
3. *Layer Internet*

   Layer internet ini berfungsi untuk melakukan pemetaan (*routing*) dan enkapsulasi *packet*. Salah satu protokol yang terlibat didalamnya adalah *Internet Protocol* (IP)
4. *Layer Network Access*

   Berfungsi untuk meletekkan *frame-frame* jaringan di atas media jaringan yang digunakan.

2.6.2 TCP dan IP

TCP dan IP merupakan protokol yang bekerja pada suatu *layer* dan menjadi penghubung antara satu komputer dengan komputer yang lain di dalam *network*, meskipun kedua komputer tersebut memiliki OS yang berbeda. Dalam melakukan

komunikasi didalam jaringan, TCP dan IP memiliki tugas masing-masing yaitu sebagai berikut:

1. TCP, *Transmission Control Protocol* merupakan *connection-oriented*, yang berarti bahwa kedua komputer yang ikut serta dalam pertukaran data harus melakukan hubungan terlebih dulu sebelum pertukaran data berlangsung. Selain itu TCP juga bertagnggungjawab untuk meyakinkan bahwa data yang dikirimkan sampai ke tujuan, memeriksa kesalahan dan mengirimkan *error* ke lapisan atas jika TCP tidak berhasil melakukan hubungan. Jika ukuran data terlalu besar untuk satu datagram, TCP akan membaginya ke dalam beberapa datagram.
2. IP, *Internet Protocol* bertanggung jawab setelah hubungan berlangsung. Tugasnya adalah untuk me-rute-kan paket data di dalam *network*. IP hanya bertugas sebagai pengirim dari TCP dan mencari jalur yang terbaik dalam penyampaian datagram. IP tidak bertanggung jawab jika data tersebut tidak sampai dengan utuh, karena IP tidak memiliki informasi mengenai isi data yang dikirimkan, namun IP akan mengirimkan pesan kesalahan (*error message*) melalui ICMP (*Internet Control Message Protocol*) jika hal ini terjadi dan kemudian kembali ke sumber data berasal.

   Karena IP hanya mengirimkan data tanpa mengetahui urutan data mana yang akan disusun berikutnya, maka hal ini menyebabkan IP mudah untuk dimodifikasi di daerah sumber dan tujuan datagram. Hal inilah yang menyebabkan adanya paket data yang hilang sebelum sampai ke tujuan [10].

2.7 *Internet Protocol version* 4 (IPv4)

IP *address* merupakan pengenal atau identifier yang digunakan untuk memberi alamat pada tiap-tiap komputer dalam jaringan. Format IPv4 address adalah 32 bit yang tiap 8 bit-nya dipisahkan oleh tanda titik, format bakunya adalah seperti demikian xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx, dengan x merupakan bilangan biner 0 atau 1. Atau dengan bentuk empat bilangan dengan titik sebagai pemisahnya seperti berikut xxx.xxx.xxx.xxx, di mana xxx merupakan nilai desimal dari 8 bit biner.

2.7.1 *Network ID* dan *Host ID*

Dalam 1 alamat IPv4 dibagi menjadi 2 bagain yaitu *network ID* dan *host ID* yang ditandai oleh *subnet mask*. Untuk menandai suatu *network ID subnet mask* di set ke angka 1 dan untuk menandai suatu *host ID subnet mask* di set ke angka 0, misalkan alamat 192.168.10.100 dengan subnet masuk 255.255.255.0 menandakan bahwa 192.168.10.0 merupakan alamat *network* dan .100 merupakan alamat *host*.

Alamat *host ID* dalam 1 *network ID* harus unik, karena *host ID* harus merupakan satu-satunya alamat yang dimiliki oleh suatu *host* dalam satu jaringan, jika terdapat 2 *host ID* dalam 1 *network ID* maka akan terjadi konflik dalam jaringan tersebut.

Alamat IPv4 dibagi menjadi 3 jenis yaitu :

1. Alamat *Unycast*

   Alamat ini digunakan dalam komunikasi *point to point.*
2. Alamat *Broadcast*

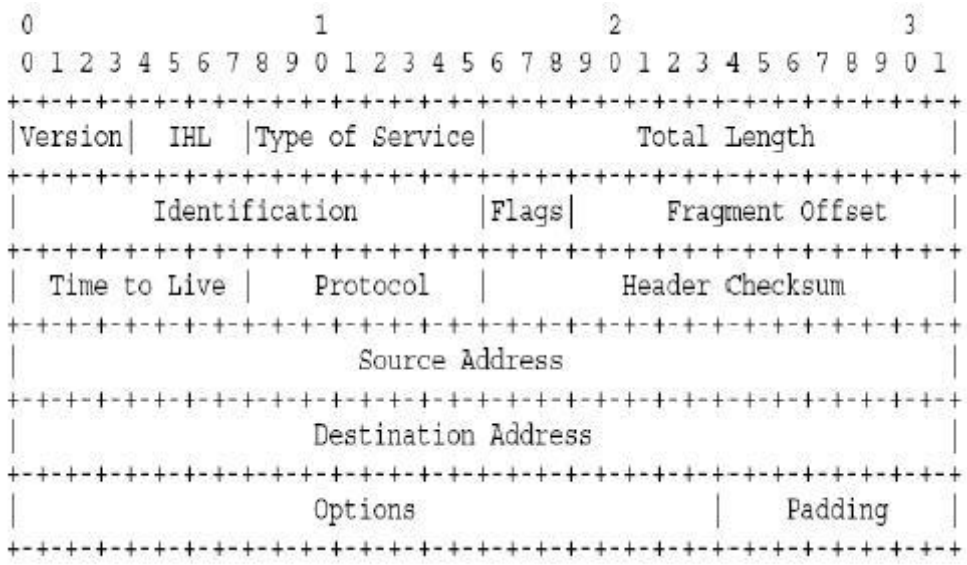
   Alamat ini digunakan dalam komunikasi *one to everyone.* Biasanya dipakai dalam satu segmen jaringan yang sama.
3. Alamat *Multicast*

   Alamat ini digunakan dalam komunikasi *one to many.*

2.7.2 Struktur *Header Internet Protocol version* 4

IPv4 dibentuk oleh struktur *header* yang memiliki 14 *field* dan panjang 32 bit. Setiap *field* memiliki fungsi dan tujuan masing-masing. Pada *header* IPv4 masih ditemukan kekurangan-kekurangan yang dapat dijadikan celah bagi penyusup atau orang-orang yang tidak bertanggungjawab untuk melakukan aksi pencurian data pada *packet* IPv4.

Berikut merupakan struktur *header* dari IPv4:



Gambar 2.5 Format *Header* IPv4

Header IPv4 terdiri atas beberapa *field*, dengan penjelasan sebagai berikut:

1. *Version*

   Digunakan sebagai tanda *version* dari *header* IP yang digunakan, yaitu IPv4.

2. *Internet Header Length*

   Digunakan untuk mengindikasikan ukuran *header* IP.

3. *Type of Service*

   Field ini digunakan untuk menentukan kualitas transmisi dari sebuah datagram IP.

4. *Total Length*

   Merupakan panjang total dari datagram IP, yang mencakup *header* IP dan muatannya.

5. *Identification*

   Digunakan untuk mengidentifikasi sebuah *packet* IP tertentu yang akan di fragmentasi.

6. *Flags*

   Berisi dua buah *flag* yang berisi apakah sebuah datagram IP mengalami fragmentasi atau tidak.

7. *Fragment Offset*

   Digunakan untuk mengidentifikasikan *offset* dimana fragmen yang bersangkutan dimulai, dihitung dari permulaan muatan IP yang belum pecah.

8. *Time to Live*

   Digunakan untuk mengidentifikasikan berapa banyak saluran jaringan dimana sebuah datagram IP dapat berjalan-jalan sebelum sebuah *router* mengabaikan datagram tersebut.

9. *Protocol*

   Digunakan untuk mengidentifikasikan jenis protokol lapisan yang lebih tinggi yang dikandung oleh muatan IP.

10. *Header Checksum*

    Field ini berguna hanya untuk melakukan pengecekan integritas terhadap *header* IP.

11. *Source IP Address*

    Field ini mengandung alamat IP dari sumber *Host* yang mengirimkan datagram IP tersebut.

12. *Destination IP Address*

    Field ini mengandung alamat IP tujuan kemana datagram IP tersebut akan disampaikan.

13. *Option*

    Fitur pendukung IP yang dapat dipilih, biasanya fitur kemanan.

14. *Padding*

    Bit-bit 0 tambahan yang ditambahkan ke dalam *field* ini untuk memastikan *header* IPv4 tetap berukuran 32 bit.

2.8 *Internet Protocol version* 6 (IPv6)

Internet Protocol version 6 atau IPv6 merupakan *internet protocol* yang dirancang untuk menggantikan IPv4. Jumlah 4.3 miliyar alamat yang dimiliki oleh IPv4 dirasa akan mengalami kekurangan untuk memenuhi jumlah kebutuhan alamat IP. IPv6 mulai dikembangkan pada 14 Juli 1999 [2]. Selain untuk memenuhi jumlah alamat yang diperlukan, IPv6 juga didesain untuk memperbaiki kekurangan-kekurangan yang ada pada IPv4.

Kekurangan pada sisi jumlah alamat IP disebabkan oleh kebutuhan manusia yang kian hari kian meningkat serta alih fungsi yang terjadi pada jaringan yang sudah menjadi kebutuhan disebagian aspek kegiatan manusia. Kebutuhan akan hiburan (*entertainment*), *channel* untuk televisi maupun *video on demand* dan sebagainya yang sudah berbasis internet memunculkan suatu kebutuhan akan hadirnya internet protokol yang mampu menangani kebutuhan-kebutuhan tersebut dengan baik.

2.8.1 Fitur IPv6

IPv6 memiliki beberapa fitur yang mampu mengantisipasi perkembangan aplikasi masa depan dan mengatasi kekurangan yang dimiliki IPv4. Fitur-fitur tersebut adalah [10]:

1. Jumlah IP address yang sangat banyak

   Sesuai dengan salah satu tujuannya, IPv6 dibuat untuk memenuhi jumlah kebutuhan alamat IP yang sangat banyak. Alamat IPv6 memiliki panjang 128 bit atau setara dengan 3.4 x 10 ^38.

2. *Autoconfiguration*

   IPv6 dirancang agar penggunanya tidak dipusingkan dengan konfigurasi IP *address*. Komputer pengguna yang terhubung dengan jaringan IPv6 akan mendapatkan IP *address* langsung dari *router* sehingga akan sangat berguna untuk peralatan *mobile* internet karena pengguna tidak direpotkan dengan konfigurasi sewaktu berpindah tempat dan jaringan.

3. *Security*

Kebutuhan *security* yang hanya menjadi optional pada IPv4 telah menjadi suatu kebutuhan pada IPv6. IPv6 telah dilengkapi dengan protokol IPSec, sehingga semua aplikasi telah memiliki sekuriti yang optimal bagi berbagai aplikasi yang membutuhkan keamanan.

4. *Quality of Services*
   IPv6 memiliki protokol QoS yang terintegrasi dengan baik sehingga semua aplikasi yang berjalan di atas IPv6 memiliki jaminan QoS.

2.8.2 Perubahan IPv4 ke IPv6

Fungsi-fungsi yang bekerja pada IPv4 juga ada pada IPv6, sedangkan fungsi-fungsi yang tidak bekerja pada IPv4 dihilangkan pada IPv6. Perubahan dari IPv4 ke IPv6 dapat dibagi dalam beberapa kategori, yaitu :

1. Kapabalitas *routing* dan pengalamatan yang semakin besar.
   Pengalamatan yang digunakan meningkat dari 32 bit pada IPv4 menjadi 128 bit pada IPv6 untuk mendukung hierarki pengalamatan dan jumlah pengalamatan node-node yang lebih banyak, dan konfigurasi alamat-alamat secara otomatis yang lebih sederhana.
2. Tipe alamat yang baru yang disebut *anycast address.*
   Didefinisikan untuk mengidentifikasi kumpulan node-node dimana paket yang dikirimkan ke sebuah *anycast address* dikirimkan ke salah satu node. Penggunaan *anycast address* pada rute sumber memungkinkan node-node mengontrol *path* tempat aliran *traffic*.
3. Penyederhanaan format *header*.
   Beberapa *field-field* pada *header* IPv4 telah dihilangkan atau lebih optional untuk mengurangi pemrosesan dari penanganan paket dan untuk menjaga *bandwidth* dari IPv6 sekecil mungkin walaupun ada peningkatan ukuran alamat.
4. Peningkatan dukungan untuk *option-option*.
   Perubahan pada penyedian *option-option* pada *header* IPv6 memungkinkan proses pelewatan paket yang lebih efisien, batasan-batasan panjang *option*

yang lebih besar, dan fleksibilitas untuk *option-option* yang mungkin ada di masa depan.

5. Kapabilitas *Quality of Services* (QoS).
   Sebuah kapabilitas baru ditambahkan untuk memungkinkan pemberian label pada paket-paket dari aliran *traffic* tertentu dimana pengirim membutuhkan penanganan khusus, seperti *quality of service* yang bukan *default* dan *service* yang *bersifat real-time*.
6. Kapabilitas authentikasi dan privasi.
   IPv6 juga mendukung authentikasi, kesatuan data, dan kerahasiaan.

2.8.3 Jenis Alamat IPv6

Pada alamat IPv6, jenis alamat *broadcast* pada IPv4 ditiadakan, namun pada IPv6 muncul sebuah pengalamatan baru dengan nama *multicast address*. Sehingga jenis alamat yang terdapat pada IPv6 sama dengan IPv4 yaitu sebanyak 3 jenis dengan penjelasan sebagai berikut [12]:

1. *Unycast address*
   Unicast address merupakan jenis IP *address* yang digunakan untuk identifikasi sebuah *interface* saja. Paket yang dikirimkan ke *unicast address* hanya akan diterima oleh sebuah *interface* yang menggunakan alamat tersebut.
2. *Anycast address*
   Anycast address merupakaan jenis IP *address* yang digunakan untuk identifikasi sekumpulan *interface*. Paket yang dikirim ke *anycast address* akan diterima oleh *interface* terdekat (salah satu *interface*) dari sekumpulan *interface* yang menggunakan alamat tersebut. Penentuan *interface* terdekat adalah berdasarkan pengukuran jarak dari protokol *routing*. *Anycast address* tidak lain adalah *unicast address* yang diberikan pada sekumpulan *interface* dengan persyaratan khusus, yaitu:
   a. *Anycast address* hanya digunakan pada *router* IPv6 saja. Tidak boleh digunakan untuk alamat *host*.

b. *Anycast address* tidak boleh digunakan sebagai alamat asal dari paket IPv6

3. *Multicast address*

   Multicast address digunakan untuk identifikasi sekumpulan *interface*. Paket yang dikirim ke *multicast address* akan diterima oleh semua *interface* yang menggunakan alamat tersebut. Secara umum, *multicast address* pada IPv6 berfungsi sama dengan multicast address pada IPv4.

2.8.4 Format Penulisan IPv6

Secara umum format penulisan IPv6 adalah x:x:x:x:x:x:x:x yang dipisahkan oleh *double-colon* (titik dua ‘:’) dan x yang merepresentasikan panjang 16 bit atau 2 octec kemudian diikuti dengan tanda slash (/) untuk menandakan prefix jaringan. Untuk menuliskan alamat IPv6, dapat dilakukan dengan cara sebagai berikut yaitu:
Alamat IPv6 : 2001:0716:0250:0000:0000:0000:0000:4/64
Dapat ditulis juga dengan bentuk:
2001:716:250:0:0:0:0:4/64 atau 2001:716:250::4/64

Tanda /64 menunjukan bahwa 64 bit pertama merupakan alamat jaringan yang digunakan. Penulisan 2 *double-colon* menunjukan bahwa bit yang ada didalamnya adalah bit 0 dan penggunaan 2 *double-colon* tersebut tidak dapat diberikan sebanyak 2 kali pada 1 alamat IPv6 karena dapat menyebabkan alamat IPv6 menjadi tidak valid.

2.8.5 Struktur *Header Internet Protocol version* 6

IPv6 memiliki beberapa fitur yang mampu mengantisipasi perkembangan aplikasi masa depan dan mengatasi kekurangan yang dimiliki pendahulunya, yaitu IPv4. IPv6 dirancang sebagai perbaikan dari IPv4. *Header* pada IPv6 terdiri dari dua jenis, yang pertama, yaitu *field* yang dibutuhkan oleh setiap paket disebut *header* dasar, sedangkan yang kedua yaitu *field* yang tidak selalu diperlukan pada packet disebut *header* ekstensi, dan *header* ini didifinisikan terpisah dari *header* dasar. *Header* dasar selalu ada pada setiap paket, sedangkan *header* tambahan hanya jika diperlukan diselipkan antara *header* dasar dengan data. *Header*

tambahan, saat ini didefinisikan selain bagi penggunaan ketika paket dipecah, juga didefinisikan bagi fungsi sekuriti dan lain-lain. *Header* tambahan ini, diletakkan setelah *header* dasar, jika dibutuhkan beberapa *header* maka *header* ini akan disambungkan berantai dimulai dari *header* dasar dan berakhir pada data. *Router* hanya perlu memproses *header* yang terkecil yang diperlukan saja, sehingga waktu pemrosesan menjadi lebih cepat. Hasil dari perbaikan ini, meskipun ukuran *header* dasar membesar dari 20 bytes menjadi 40 bytes namun jumlah *field* berkurang dari 12 menjadi 8 buah saja.

Berikut ini merupakan format *header* dari IPv6.



Gambar 2.6 Format *Header* IPv6

Field-field pada *header* IPv6 dapat dijelaskan secara singkat sebagai berikut:

1. *Version*

   Field 4 bit yang menunjukkan versi *Internet Protocol*, yaitu 6.

2. *Prior*

   Field 4 bit yang menunjukkan nilai prioritas. *Field* ini memungkinkan pengirim paket mengidentifikasi prioritas yang diinginkan untuk paket yang dikirimkan, relatif terhadap paket-paket lain dari pengirim yang sama.

3. *Flow Label*

   Field 24 bit yang digunakan oleh pengirim untuk memberi label pada paket-paket yang membutuhkan penanganan khusus dari *router* IPv6,

seperti *quality of service* yang bukan default, misalnya *service-service* yang bersifat *real-time.*

4. *Payload Length*

   Field berisi 16 bit yang menunjukkan panjang *payload*, yaitu sisa paket yang mengikuti *header* IPv6, dalam oktet.

5. *Next Header*

   Field 8 bit yang berfungsi mengidentifikasi *header*

   Berikut ini merupakan yang mengikuti *header* IPv6 utama.

6. *Hop Limit*

   Field berisi 8 bit *unsigned integer*. Menunjukkan jumlah *link* maksimum yang akan dilewati paket sebelum dibuang. Paket akan dibuang bila *Hop Limit* berharga nol.

7. *Source Address*

   Field 128 bit, menunjukkan alamat pengirim paket.

8. *Destination Address*

   Field 128 bit, menunjukkan alamat penerima paket.

2.9 Migration Plan

Migrasi merupakan suatu kata yang cukup familiar dibanyak hal. Di dunia hewan, migrasi biasa dilakukan oleh kawanan hewan untuk berpindah tempat dari satu tempat ke tempat yang lain. Alasan mendasar dari perpindahan tersebut adalah suber daya alam yang tersedia di tempat semula sudah tidak dapat mecukupi kawanan hewan tersebut untuk tetap dapat bertahan hidup.

Langkah migrasi yang diputuskan oleh kawanan hewan tersebut bukan tanpa resiko. Resiko utama dalam melakukan migrasi adalah pemangsa yang siap menggangu kawanan hewan tersebut dalam perjalanan menuju tempat tinggal yang baru. Namun resiko tersebut lebih kecil dibanding resiko kelaparan karena sumber daya alam yang sudah tidak mencukupi. Sehingga langkah migrasi dilakukan oleh kawanan hewan tersebut untuk menjamin kelangsungan hidup mereka.

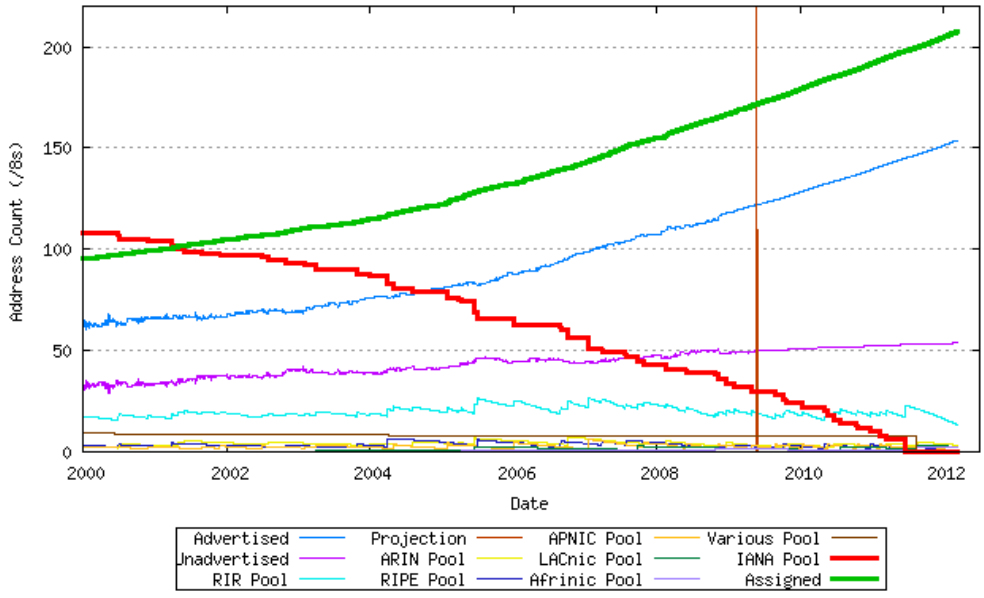
Senada dengan dunia hewan, migrasi pun terjadi di dunia jaringan. *Internet protocol* yang selama ini berbasis IPv4 sudah mulai beralih ke IPv6, ketersediaan

IPv4 sudah tidak memungkinkan lagi untuk dapat memenuhi kebutuhan yang diperlukan manusia dalam hal pengembangan serta eksplorasi dan ekploitasi ilmu pengetahuan dan teknologi yang saat ini sudah berbasis jaringan yang saling terhubung satu sama lainnya atau berbasis internet.

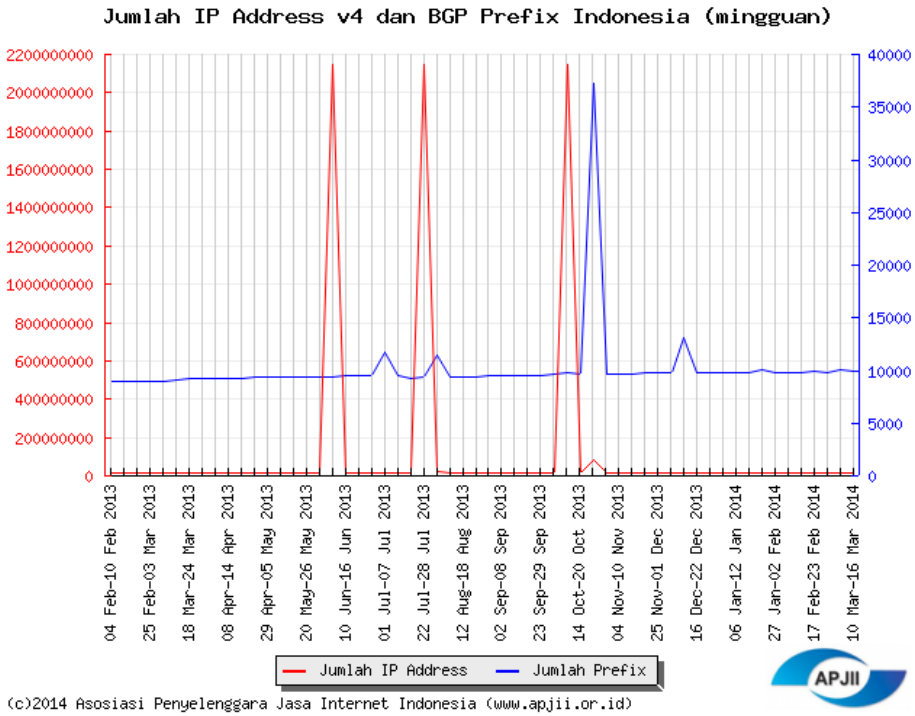
Saat ini jumlah penduduk dunia mencapai 7 milyar, dan penetrasi internet berkisar 2 milyar pengguna. Jika setiap pengguna memiliki 1 buah perangkat yang terkoneksi ke internet, berarti 50% dari total alamat IPv4 sudah habis terpakai. Nyatanya, setiap orang saat ini bisa memiliki beragam *gadget*, komputer di kantor, komputer di rumah, laptop, beberapa *smartphone* dan perangkat-perangkat lainnya yang berbasis IP. Belum lagi kebutuhan IP *address* pada infrastruktur seperti *router*, *bandwith management*, *server*, *firewall*, *access point*, *switch* dll. Maka sudah sangat wajar jika alamat IPv4 yang tersedia sudah sangat tipis jumlahnya [13].

Dari hasil kalkulasi matematis, banyak pakar jaringan memprediksikan bahwa *free pool* IPv4 di IANA akan habis pada pertengahan tahun 2011. *Internet Assigned Numbers Authority* (IANA) merupakan departemen internasional yang bertanggung jawab untuk mengkoordinasikan *DNS Root*, *IP Addressing* dan *Internet Protocol Resource* diseluruh dunia. [2]

Berikut ini tampilan grafik penggunaan IPv4 didunia sampai akhir tahun 2012 dan penggunaan IPv4 pada tahun 2014:



Gambar 2.7 Grafik Penggunaan IPv4 di Dunia [13]



Gambar 2.8 Jumlah IP Address v4 dan BGP Prefix Indonesia [13]

Dengan keadaan seperti ini muncul kekhawatiran terhadap perkembangan internet. Di sisi penyedia jaringan, habisnya IPv4 akan menghambat pertumbuhan pengguna internet baru. Pengguna yang sebelumnnya menggunakan IP *address* publik, terpaksa harus menggunakan IP *address privat*. Hal ini akan berakibat adanya keterbatasan akses, dan tidak bisa menikmati aplikasi/layanan yang menuntut adanya *end-to-end connectivity*, karena mesin yang memiliki IP *privat*, hanya dapat mengakses *internet*, tetapi tidak dapat diakses langsung dari *internet* (komunikasi satu arah).

Untuk mengatasi hal tersebut, migrasi ke IPv6 merupakan satu-satunya cara yang dapat menjaga kelangsungan pertumbuhan *internet*. IPv6 memungkinkan pengalamatan yang jauh lebih besar dengan kapasitas 2 pangkat 128 alamat. Selama masa transisi, IPv6 dan IPv4 akan dijalankan bersama-sama dan dikenal dengan nama *Dual Stack.*

Pengguna yang masih hanya menggunakan IPv4 saja tanpa IPv6 akan mengalami kesulitan untuk mengakses situs atau layanan yang hanya berbasis IPv6,

begitu pula sebaliknya. Karena Jaringan IPv6 dan jaringan IPv4 tidak bisa saling berhubungan. [13]

2.9.1 Istilah-Istilah Node Pada Mekanisme Transisi

Istilah-istilah node yang digunakan dalam mekanisme transisi dijelaskan dalam RFC4213 "*Basic Transition Mechanism for IPv6 Hosts and Routers*" yang dikeluarkan oleh IETF (*Internet Engineering Task Force*), yaitu sebagai berikut:

1. *IPv4-only node*: *host* atau *router* yang hanya mendukung IPv4 dan tidak mendukung IPv6.
2. *IPv6/IPv4 node*: *host* atau *router* yang mendukung IPv6 dan IPv4. Node ini disebut *dual stack node*.
3. *IPv6-only node*: *host* atau *router* yang hanya mendukung IPv6 dan tidak mendukung IPv4.
4. *IPv4 node*: *host* atau *router* yang mendukung IPv4. *Host IPv6/IPv4* dan *IPv4-only node* termasuk didalamnya.
5. *IPv6 node*: *host* atau *router* yang mendukung IPv6. *Host IPv6/IPv4* dan *IPv6-only* node termasuk didalamnya. [3]

2.9.2 Strategi Migration Plan

Untuk melakukan migrasi jaringan terhadap jaringan IPv6 diperlukan suatu kajian tertentu yang memungkinkan langkah migrasi tersebut dilakukan. Untuk mengimplemetasikan IPv6 sebagai protokol internet dibutuhkan banyak waktu dan kesiapan terutama kesiapan akan infrastruktur serta *operational* jaringan sebagai *administrator* yang mampu menangani permasalahan-permasalahan yang muncul ketika IPv6 mulai diimplementasikan didalam lingkungan jaringannya.

Dalam beberapa informasi yang didapat penulis, setidaknya dibutuhkan waktu sekitar 10 sampai 15 tahun bagi manusia untuk dapat mengimplementasikan IPv6 sebagai internet protokol yang digunakan secara luas dan umum diseluruh dunia.

Memang saat ini tidak nampak secara nyata bagaimana IPv6 diimplementasikan dalam jaringan internet. Dan bagi sebagian orang bahkan

negara, hal tersebut dapat diabaikan karena tertutup oleh permasalahan dan kebutuhan lainnya. Namun dimasa yang kompetitif ini orang-orang yang jeli akan melihat langkah migrasi IP ini sebagai salah satu cara agar suatu lingkungan jaringan secara kompetitif dapat *reliable* dengan jaringan IPv4 maupun IPv6. Sehingga percepatan pembangunan infrastruktur jaringan dapat dilakukan untuk menyesuaikan kebutuhan terhadap pertumbuhan dan perkembangan teknologi yang sangat pesat.

Dalam melakukan migrasi IP terdapat beberapa teknik atau metode yang dapat diimplemetasikan sesuai dengan kebutuhan dan keadaan lingkungan jaringan yang sudah ada. Masing-masing metode memiliki karakteristik yang dapat dijadikan pertimbangan untuk dipilih sebagai cara dalam melakukan migrasi IP.

Terdapat 3 teknik dasar dalam melakukan migrasi IP, yaitu dengan mekanisme *Dual-Stack* dimana IPv4 maupun IPv6 dapat dioperasikan secara bersama-sama kemudian mekanisme *tunneling* dimana packet IPv6 dibungkus oleh header IPv4 untuk dapat dilewatkan pada jaringan IPv4 dan yang ketiga adalah mekanisme *translation* dimana IPv4 maupun IPv6 diubah sesuai dengan alamat *host* tujuan dengan melewati perangkat jaringan yang memiliki kemampuan *Dual-Stack.*

2.9.2.1 Dual-Stack

Dual-Stack merupakan strategi migrasi dengan menggunakan protokol IPv6 dan IPv4 dalam satu infrastruktur jaringan. Infrastruktur tersebut dapat berupa *host* komputer maupun perangkat jaringan lainnya seperti *router*. Sehingga *host* atau lingkungan jaringan yang menerapkan mekanisme *Dual-Stack* dapat menjalankan kedua protokol IPv6 dan IPv4 secara paralel sesuai dengan kebutuhan yang digunakan saat komunikasi berlangsung.

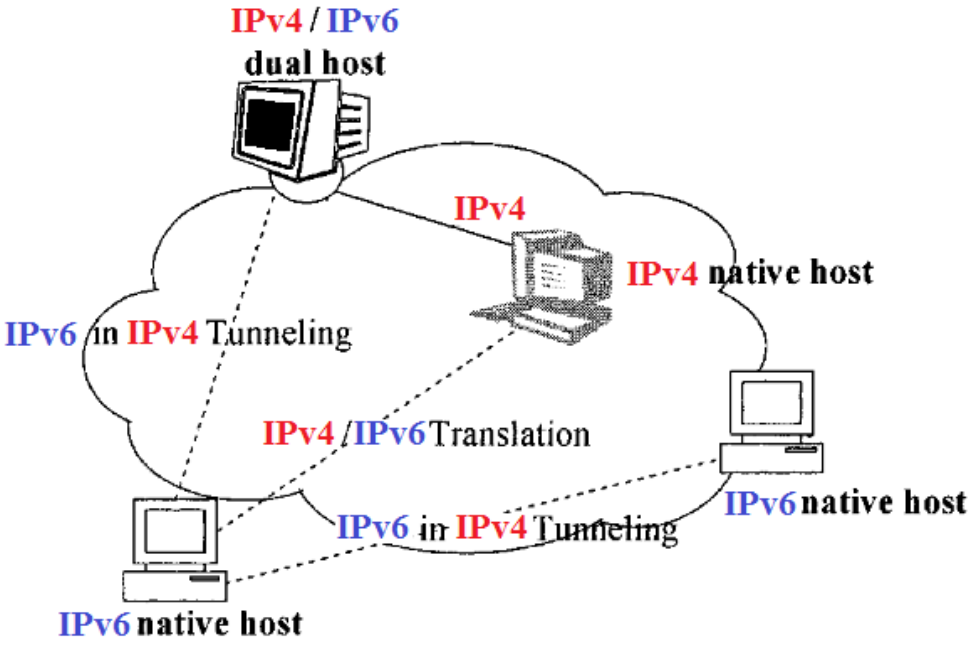
Sebagai contoh, misalkan *node Dual Stack* berkomunikasi dengan *IPv4-only node*, maka *node Dual Stack* tersebut akan menggunakan protokol IPv4 sebagai jalur komunikasinya dan sebaliknya misalkan *node Dual Stack* tersebut berkomunikasi dengan *IPv6-only node*, maka *node Dual Stack* tersebut akan menggunakan protokol IPv6 sebagai jalur komunikasinya. Namun jika kedua node

tersebut sama-sama menerapkan mekanisme *dual stack* dan memiliki IPv6 maupun IPv4 maka komunikasi akan berlangsung berdasarkan protokol *preferred* yang dioperasikan.

Walaupun kedua protokol digunakan secara bersama-sama, namun salah satu dari kedua protokol tersebut dapat di *disable*-kan karena suatu alasan tertentu. Terdapat 3 kemungkinan yang dapat dioperasikan terhadap *node Dual Stack*, yaitu:

1. *IPv4 stack enabled* dan *IPv6 stack disabled*.
2. *IPv6 stack enabled* dan *IPv4 stack disabled*.
3. *IPv4* dan *IPv6 stack enable* kedua-duanya. [14]

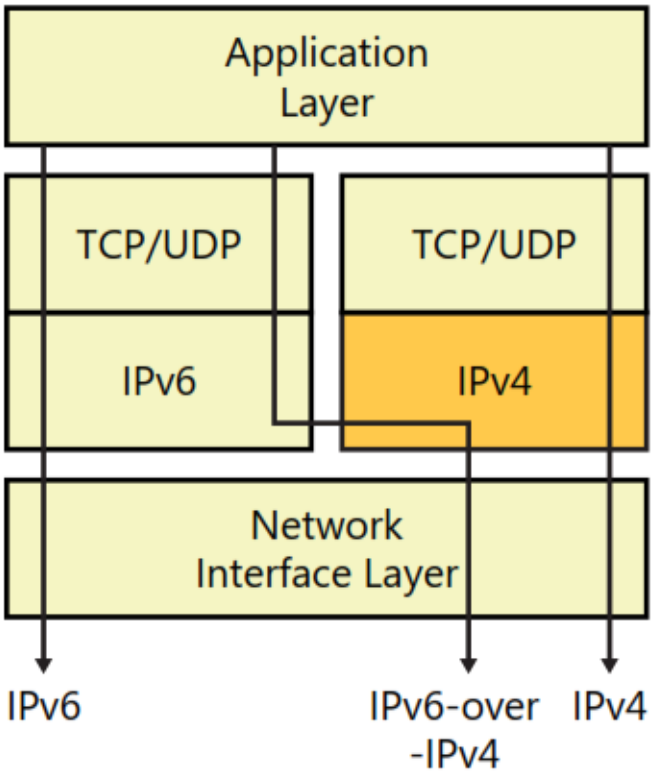
Dual Stack merupakan konsep transisi yang paling biasa digunakan, karena memungkinkan untuk kedua protokol berjalan masing-masing tanpa saling mempengaruhi. *Dual Stack* juga merupakan kemampuan dasar yang dimiliki oleh perangkat jaringan yang dapat beroperasi dengan IPv4 maupun IPv6.



Gambar 2.9 Skenario Jaringan *Dual Stack* IPv4 dan IPv6 [14]

2.9.2.1.1 *Dual Stack Architecture*

Arsitektur *Dual-Stack* terdiri dari IPv4 dan IPv6 *internet layers*, tetapi keduanya berada pada *protocol stacks* yang berbeda yang berisi implementasi terpisah dari *Transport layer protokol* seperti TCP dan UDP. [3]



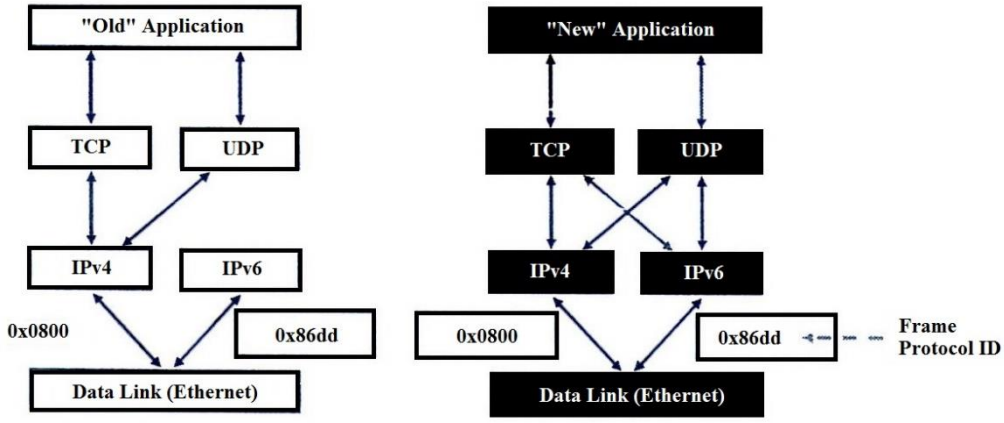
Gambar 2.10 Arsitektur *Dual Stack* [3]

2.9.2.1.2 *Dual Stack Applications*

Mekanisme *Dual-Stack* tidak hanya diimplemntasikan pada perangkat-perangkat jaringan. Di sisi aplikasi, penerapan mekanisme *Dual Stack* menjadi yang lebih diminati karena memiliki 2 protokol IPv4 dan IPv6 dalam satu *node* yang sama.

Pada pengembangan aplikasi-aplikasi terbaru, perubahan serta modifikasi dibutuhkan agar aplikasi dapat berjalan diatas protokol jaringan IPv6 sementara untuk aplikasi-aplikasi yang lama masih tetap dapat berjalan dan berkomunikasi diatas protokol jaringan IPv4.

Application Programming Interface (API) didefinisikan untuk dapat mendukung pengalamatan dan *Domain Name Services request* IPv4 dan IPv6. API terbaru ini mengganti yang disebut ***gethostnyname*** dan ***gethostbyaddr*** pada aplikasi. Perubahan API ini membuat aplikasi dapat berjalan diatas protokol IPv4 maupun IPv6 dan selama proses peralihan tersebut aplikasi tetap dapat berjalan diatas protokol jaringan IPv4 [15].



Gambar 2.11 *Dual Stack Applications* [15]

2.9.2.1.3 Proses Pemilihan Alamat *Host Dual Stack*

Pengaturan pemilihan alamat dan DNS *resolution* pada *host Dual-Stacking* dilakukan dengan proses sebagai berikut:

1. Step 1: DNS *request* terkirim

   Dual Stack apllication mengirim DNS *request* dengan menggunakan kedua jenis protokol ke *destination hostname*, contoh *hostname* tujuan: http://www.example.com.
2. Step 2: DNS *server* mengirim pesan balasan

   DNS *server* mengirim pesan balasan kesemua alamat *host* yang tersedia, dengan menggunakan IPv6 dan IPv4.
3. Step 3: Memilih alamat yang sesuai.

   Aplikasi memilih alamat yang digunakan secara *default*, contoh menggunakan IPv6.

4. Step 4: Aplikasi terhubung dengan *host* tujuan
   Aplikasi sebagai alamat *source* terhubung dengan *destination host* dengan menggunakan alamat IPv6. [15]

2.9.2.1.4 Langkah-Langkah Mengimplementasikan *Dual Stack*

Berikut adalah langkah-langkah yang dibutuhkan saat akan mengimplementasikan mekanisme *Dual Stack*:

1. Me-*review* jaringan, aplikasi dan keamanan *network environment*.
2. Meng-*upgrade host*, *router* dan *infrastructure services* untuk mendukung IPv6 jika dibutuhkan.
3. Memungkinkan *support* IPv6.
4. Meng-*upgrade* semua servis yang memungkinkan dapat menyediakan layanan IPv6.
5. Memastikan bahwa implemntasi *Dual Stack* telah benar dan berhasil dioperasikan [15].

2.9.2.1.5 Kelebihan dan Kekurangan *Dual Stack*

Kelebihan penerapan mekanisme *Dual Stack*:

1. Relatif mudah diimplementasikan karena dikonfigurasi langsung pada perangkat internal sistem jaringan yaitu *router*.
2. Biaya yang dikeluarkan cukup besar diawal namun manjadi investasi untuk masa mendatang karena spesifikasi dari perangkat yang digunakan sudah mendukung implementasi jaringan IPv6, sehingga biaya yang dikeluarkan dapat digunakan untuk jangka waktu yang lama.
3. Dapat mengembangkan infrastruktur IPv4 yang sudah ada.
4. Memberikan kesinambungan terhadap keberadaan IPv4 dan IPv6 secara bersama-sama.

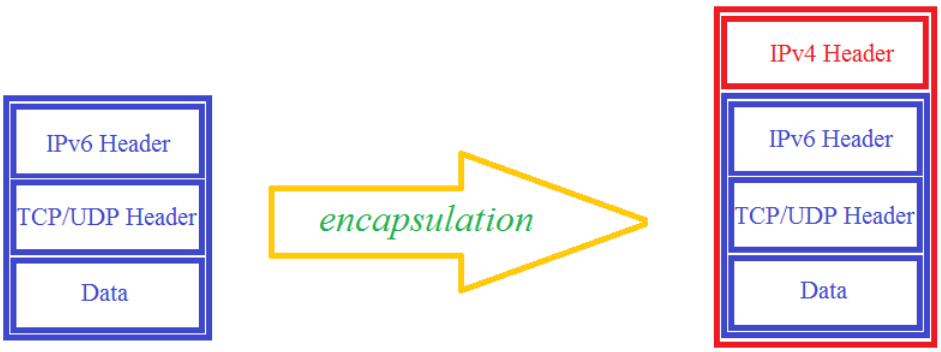
Kekurangan penerapan mekanisme *Dual Stack*:

1. Dapat menyebabkan pengeluaran biaya yang mahal pada jangka pendek.
2. Dapat meningkatkan kompleksitas jaringan. [15]

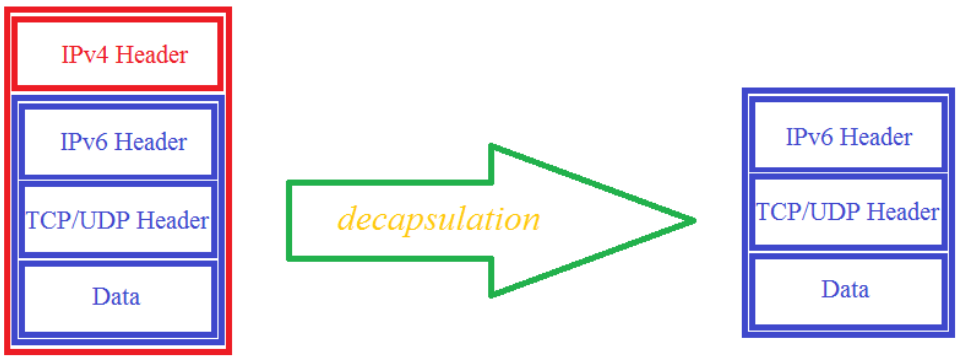
2.9.2.2 IPv6 over IPv4 Tunnel

Mekanisme *tunneling* merupakan suatu konsep migrasi dimana *packet* dengan *header* IPv6 akan ‘dibungkus’ dengan *header* IPv4 untuk dilewatkan pada jaringan IPv4. Ada banyak jenis mekanisme *tunneling* yang dapat digunakan untuk melakukan migrasi IP, yaitu:

1. *IPv6 configure tunnel*
2. *Automatic tunnel with IPv4-compatible IPv6 address*
3. *6over4 tunnel*
4. *6to4 tunnel*
5. ISATAP
6. DSTM
7. *Tunnel Broker*
8. *Teredo*



Gambar 2.12 Enkapsulasi *Packet* IPv6 menjadi *Packet* IPv4 [14]



Gambar 2.13 Dekapsulasi *Packet* IPv4 menjadi *Packet* IPv6 [14]

Mekanisme *tunneling* secara umum telihat pada gambar 2.12 dan gambar 2.13 dan prosedur tunneling secara umum dilakukan sebagai berikut [14]:

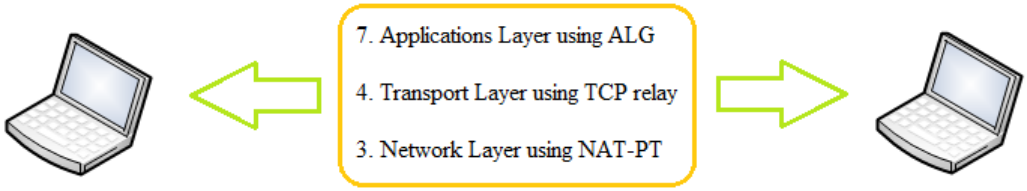
1. *Tunnel entrance node* (enkapsulator) membangun *header* IPv4, kemudian membungkus *packet* IPv6 dengan *header* IPv4 kemudian melewakan *packet* tersebut ke dalam jaringan IPv4.
2. *Tunnel exit node* (dekapsulator) menerima *packet* kemudian membuka *header packet* IPv4 agar *packet* kembali menjadi *packet* IPv6

Dengan melakukan *tunneling*, proses *routing* pada jaringan akan tetap menggunakan proses *routing* pada packet IPv4 sehingga kelebihan IPv6 yang memiliki kecepatan *routing* yang lebih baik dibandingkan IPv4 tidak dapat dirasakan. Namun hal tersebut tetap membuat *packet* dengan *header* IPv6 sampai ke tujuan dengan alamat IPv6 walaupun melewati jaringan IPv4.

2.9.2.3 *Translation Mechanism*

Konektifitas antara IPv4 dengan IPv6 dapat dilakukan dengan cara translasi atau penerjemahan. Sehingga *packet* IPv4 akan diterjemahkan ke *packet* IPv6 jika berkomunikasi dengan *host* IPv6 dan *packet* IPv6 akan diterjemahkan ke *packet* IPv4 jika berkomunikasi dengan *host* IPv4.

Dalam proses translasi, terdapat 3 *layer* yang dapat digunakan sebagai jalan untuk melakukan translasi, ketiga *layer* tersebut dapat dilihat pada gambar berikut:



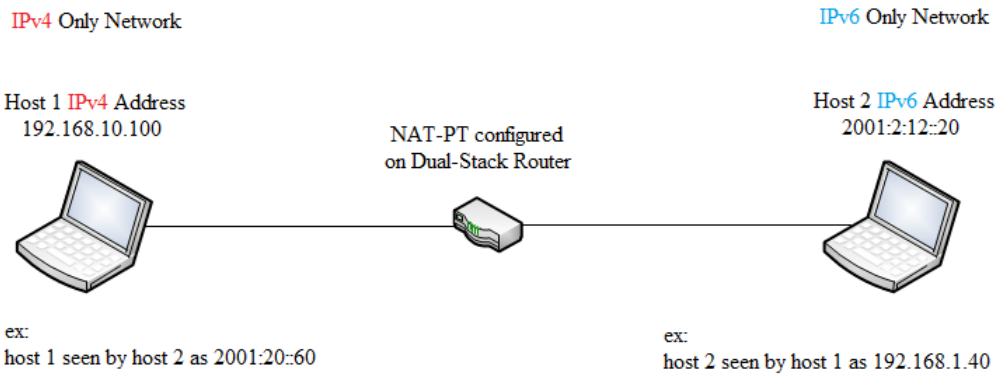
Gambar 2.14 *Translations Layers* [16]

Pada gambar 2.14 dapat dilihat bahwa mekanisme translasi dapat dilakukan pada *layer Network* dengan menggunakan NAT-PT (*Network Address Translation-Protocol Translation*), pada *layer Transport* dapat menggunakan TCP (*Transmission Control Protocol*) *relay* dan pada *layer Application* dapat menggunakan ALG (*Application Layer Gateway*).

Meskipun mekanisme translasi merupakan salah satu teknik migrasi IP namun mekanisme translasi merupakan pilihan terakhir yang dilakukan sebagai teknik migrasi IP. Hal tersebut karena terdapat banyak kompleksitas yang ditimbulkan saat proses translasi berlangsung dan tidak semua informasi, baik yang berasal dari IPv4 ke IPv6 ataupun sebaliknya dapat ditranslasikan.

2.9.2.3.1 *Network Address Translation-Protocol Translation*

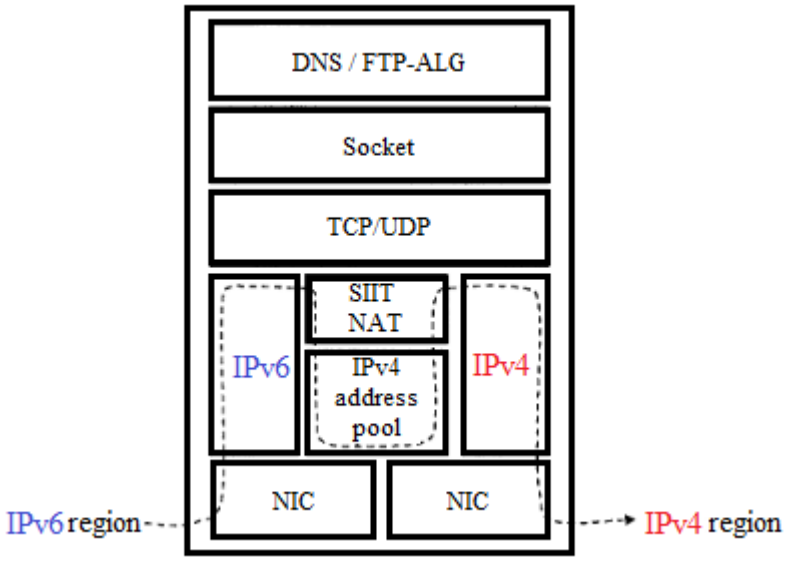
Network Address Translation-Protocol Translation (NAT-PT) merupakan salah satu mekanisme translasi yang dilakukan pada *layer Network*. Dengan menggunakan NAT-PT *host* yang terhubung akan mengenali *host* tujuan dengan versi internet protokol yang sama, walaupun sebenarnya berbeda.



Gambar 2.15 Skema NAT-PT

Pada gambar 2.15 terdapat 2 buah *host* yang memiliki alamat jaringan berbeda versi, *host* 1 menggunakan IPv4 dan *host* 2 menggunakan IPv6 yang terhubung melalui sebuah *router* yang telah di konfigurasi NAT-PT. Jika *host* 1 ingin berkomunikasi dengan *host* 2 maka alamat *host* 2 dikenali sebagai 192.168.1.40 padahal *host* 2 dikonfigurasi dengan menggunakan IPv6, begitu pun sebaliknya jika *host* 2 akan berkomunikasi dengan *host* 1 maka *packet* akan ditujukan ke alamat 2001:20::60.

2.9.2.3.2 Arsitektur NAT-PT



Gambar 2.16 Arsitektur NAT-PT [14]

Gambar 2.16 menunjukan arsitektur dari NAT-PT. Proses yang berlangsung dalam NAT-PT adalah sebagai berikut [14]:

1. Ketika *packet* diterima dari IPv6 domain, proses *filtering* digunakan untuk memeriksa apakah alamat tersebut valid atau tidak.
2. Setelah alamat diverifikasi, NAT-PT akan melihat kedalam IPv4 *address pool* yang ditetapkan sebagai alamat IPv4 dari alamat IPv6 yang diterima.

Proses tersebut membuat perbedaan *internet protocol* tidak terlihat, karena host pengirim dan penerima seperti memiliki *internet protocol* yang sama.

{Halaman ini dikosongkan}