

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 *Internet Protocol (IP)*

IP merupakan salah satu protokol yang terdapat pada *Internet Layer*. Pada saat ini, ada dua macam *Internet Protocol* yang digunakan, yaitu IPv4 dan IPv6.

##### 2.1.1 *IPv4 (Internet Protocol version 4)*

Protokol layer *network* dalam rangkaian protokol TCP/IP yang digunakan saat ini adalah IPv4. IPv4 dikembangkan pada awal tahun 1980-an dengan diperkenalkannya RFC 791, telah digunakan selama dua puluh tahun dan terbukti tangguh selama ini. IPv4 menyediakan komunikasi *host-to-host* antar sistem dalam internet. Meskipun IPv4 dirancang dengan baik, namun tetap mempunyai kekurangan sehingga tidak cocok lagi digunakan untuk internet yang berkembang secara cepat.

##### 2.1.2 *IPv6 (Internet Protocol version 6)*

IPv6 (*Internet Protocol version 6*) dirancang oleh IETF untuk mengatasi kekurangan-kekurangan yang terdapat pada IPv4. IPv6 juga dikenal sebagai IPng (*Internet Protocol next generation*). Perkembangan IPv6 dimulai pada tahun 1991. Inti dari IPv6 telah distandarisasi pada tahun 1995 dan diperbaharui dalam RFC 2460 tahun 1998. Dalam IPv6, *Internet Protocol* dimodifikasi secara luas untuk mengakomodasi pertumbuhan internet yang tidak dapat diprediksi. Format dan panjang alamat IP diubah, begitu juga format paketnya.

##### 2.1.2.1 **Kelebihan IPv6**

Masalah keterbatasan alamat pada IPv4 dapat diselesaikan dengan adanya IPv6 yang alamatnya terdiri dari 128 bit sehingga dapat memberikan alamat sejumlah  $2^{128}$ . Selain itu, IPv6 mempunyai beberapa kelebihan dan layanan tambahan, antara lain:

1. Memiliki *header* yang lebih sederhana dan efisien untuk mengurangi beban penanganan paket dan memudahkan perluasan paket oleh *router*.
2. Adanya mekanisme konfigurasi alamat secara otomatis (*address auto configuration*) oleh setiap *node*.

3. Mempunyai kemampuan *flow labeling* yaitu kemampuan untuk menandai paket yang membutuhkan penanganan khusus seperti layanan *real time*.
4. Mendukung sistem autentifikasi untuk kerahasiaan data pada lapis IP (*IP Secure*).
5. Mampu berinteraksi dengan IPv4, sehingga perubahan dari IPv4 ke IPv6 dapat dilakukan secara bertahap.
6. Mendukung *mobile IP*.

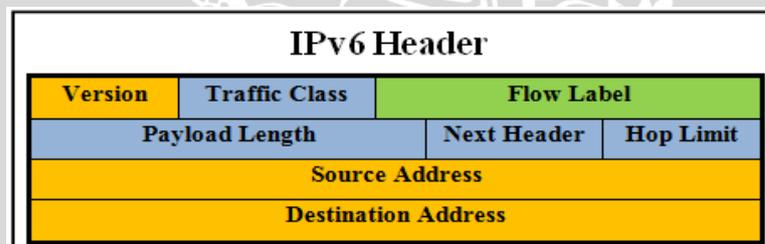
Adanya IPv6 tidak membuang secara keseluruhan fungsi IPv4 pada jaringan, melainkan ada suatu mekanisme transisi tertentu yang memungkinkan kedua versi IP tersebut berkomunikasi.

### 2.1.2.2 Arsitektur IPv6

Header IPv6 didesain mempunyai *field* yang lebih sedikit dibandingkan dengan IPv4 yaitu sebesar 40 *byte*. Hal ini menyebabkan proses penjaluran atau aliran data menjadi lebih ringkas karena penanganan yang dilakukan oleh *router* terhadap tiap *field* menjadi lebih sedikit.

Secara garis besar, paket IPv6 dibagi menjadi 2 jenis *header* yaitu:

1. *Base Header* yaitu *header* yang selalu diberikan pada paket IPv6.



**Gambar 2.1** Format *base header* (*header* dasar)/*header* IPv6  
Sumber: Taufan Riza (2002:51)

#### *Version*

*Version* adalah bit-bit yang ditambahkan oleh *header* IP untuk tiap data yang melewati protokol IP dan pada data tersebut diberikan informasi tentang versi IP yang digunakan. Dalam IPv6, *field* ini berisi 4 bit (0110 dalam biner) yang bernilai 6.

**Traffic Class (kelas lalu lintas) / Different Service**

*Field* ini mempunyai fungsi yang hampir sama dengan *field type of service* pada *header IPv4*, yaitu untuk menentukan dan membedakan kelas-kelas atau prioritas yang berlainan dari paket IPv6. *Field* ini berukuran 8 bit.

**Flow Label**

Untuk menandai aliran data tertentu yang membutuhkan penanganan khusus pada paket IPv6. Penanganan khusus tersebut dilakukan oleh *router* yang dilalui paket tersebut di dalam jaringan. Contohnya adalah pada penggunaan aplikasi tertentu misalnya aplikasi *real time*.

**Payload Length**

Digunakan untuk menandai panjang *payload*. *Payload* adalah sisa paket setelah *header IPv6* dalam bentuk oktet (di luar *header IPv6*). *Header* tambahan yang ada dihitung sebagai bagian dari *payload*, termasuk panjang *header* tambahan tersebut.

**Next Header**

Merupakan *field* 8 bit yang menentukan *header* yang mengikuti *base header* dalam *datagram*. *Header* ini merupakan salah satu pilihan *extension header* yang digunakan oleh IP atau *header upper-layer protocol* seperti TCP atau UDP. Masing – masing *extension header* juga memuat *field* ini.

**Hop Limit**

Untuk menandai maksimum *hop* yang dapat digunakan oleh IPv6 dalam lalu lintas internet.

**Source Address**

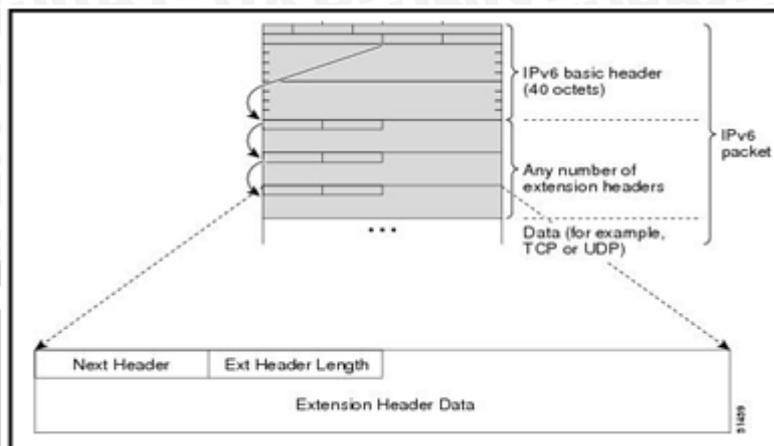
Digunakan untuk menyimpan alamat IPv6 dari *host* asal. Ukuran *field* ini 128 bit.

**Destination Address**

Digunakan untuk menyimpan alamat IPv6 dari *host* tujuan. Ukuran *field* ini 128 bit.

2. *Extension Header* (*header* tambahan/pilihan) yaitu *header* pilihan yang tidak selalu diberikan pada paket IPv6, diletakkan antara *base header IPv6* dan *header protokol upper layer* (TCP/UDP). Masing – masing *extension header*

panjangnya 8 byte (64 bit). Seluruh *extension header* secara bersama-sama membentuk daftar rangkaian *header*. *Extension header* tersedia untuk sejumlah fungsi yang berbeda, tetapi tidak dianggap sebagai bagian *header* IPv6 itu sendiri.



**Gambar 2.2** Format *extension header* IPv6

**Sumber:** Taufan Riza (2002:53)

*Extension header* meliputi:

#### ***Hop by Hop Header***

*Header* ini membawa informasi tambahan (bila ada) serta harus diproses oleh tiap *node* dan *router* sepanjang jalur pengiriman. *Header* ini terletak tepat setelah IPv6 *base header*.

#### ***Destination Option Header***

Membawa informasi pilihan yang secara khusus ditujukan untuk alamat tujuan paket.

#### ***Routing Header***

*Routing header* mendorong proses *routing* melalui sejumlah *intermediate router*. *Header* ini berisi daftar *node* yang harus dilewati dalam perjalanan menuju tujuan. Masing-masing *node* dalam daftar memproses *header* dan mengubah alamat tujuan ke *router* selanjutnya dalam daftar. *Header* ini serupa dengan *option loose source route* pada IPv4.

#### ***Fragment Header***

Digunakan oleh sumber ketika paket difragmentasi, paket dengan ukuran lebih besar akan dipecah sesuai dengan *path* MTU (*Maximum Transmission Unit*). Fragmentasi hanya dilakukan oleh sumber paket,

bukan *router*. Posisi fragmen tertentu dalam paket asli dan digunakan oleh *node* tujuan untuk menyusun kembali paket.

#### **IPSec Authentication Header**

Menyediakan kerahasiaan, integritas, dan autentikasi sumber.

#### **IPSec ESP Header**

Menyediakan layanan yang sama seperti *authentication header*, tetapi menambahkan fungsi *enkripsi* (proses perubahan data dalam bentuk sandi untuk keamanan informasi).

### **2.1.2.3 Maximum Transmission Unit (MTU) IPv6**

*Maximum Transmission Unit* IPv6 adalah ukuran maksimum paket IPv6 sebelum menuju lapisan *interface*. Besarnya MTU IPv6 tidak kurang dari 1280 byte dan dapat lebih besar tergantung jenis lapisan *interface* yang digunakan. Misalnya jika lapisan *interface* menggunakan *Ethernet* maka besar MTU adalah 1500 *byte*.

**Tabel 2.1** MTU IPv6 untuk teknologi LAN atau WAN

<b>Teknologi LAN atau WAN</b>	<b>MTU IPv6 (byte)</b>
Ethernet (Ethernet II encapsulation)	1500
Ethernet (IEEE 802.3 SubNetwork Access Protocol [SNAP] encapsulation)	1492
Token Ring	Bervariasi
FDDI	4352
Attached Resource Computer Network (ARCNet)	9072
PPP	1500
X.25	1280
Frame Relay	1592
Asynchronous Transfer Mode (ATM) – (Null or SNAP encapsulation)	9180

**Sumber:** Joseph Davies (2003:119)

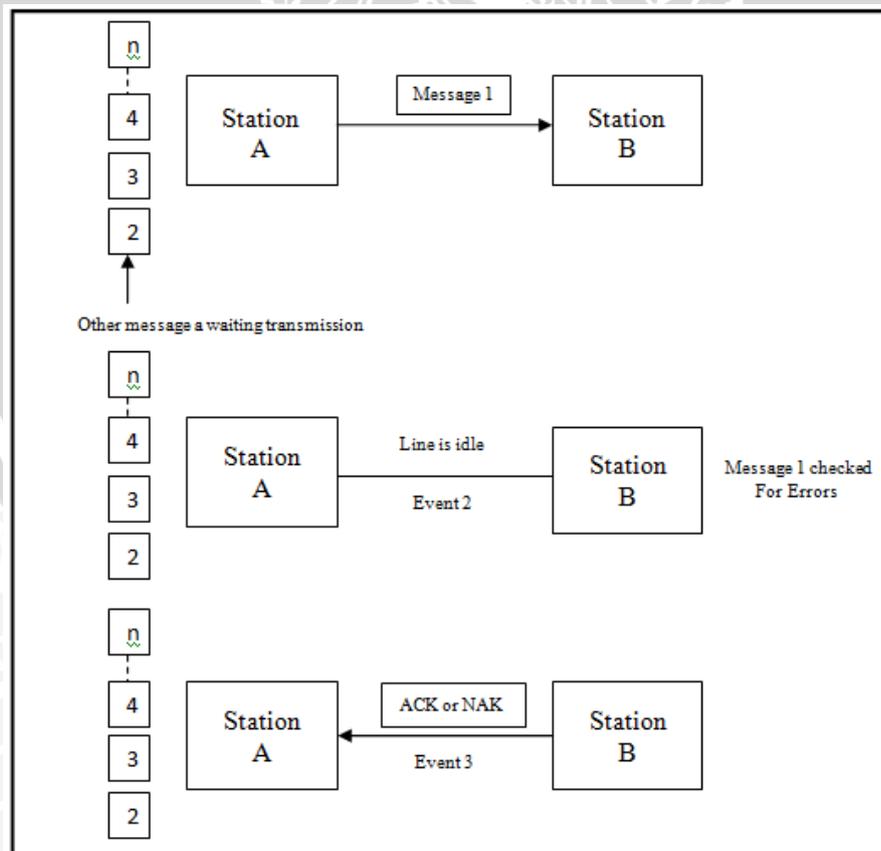
## **2.2 Konsep Dasar HSUPA**

HSUPA (*High Speed Uplink Packet Access*) didasarkan pada standar 3GPP (*3<sup>rd</sup> Generation Partnership Project*) Release 6 dan merupakan pengembangan dari jaringan WCDMA HSDPA. Pengembangan dimaksudkan

untuk mengimbangi kemampuan *downlink* HSDPA yang mencapai 14,4 Mbps karena HSDPA hanya memiliki kemampuan *uplink* sebesar 384 Kbps. Oleh karenanya diimbangi dengan kemampuan *uplink* HSUPA sebesar 5,76 Mbps. Konsep dasar HSUPA terdiri dari tiga komponen antara lain HARQ (*Hybrid Automatic Repeat Request*), *Node B Controlled Scheduling* dan TTI (*Transmission Time Interval*) yang lebih pendek.

**2.2.1 HARQ (*Hybrid Automatic Repeat Request*)**

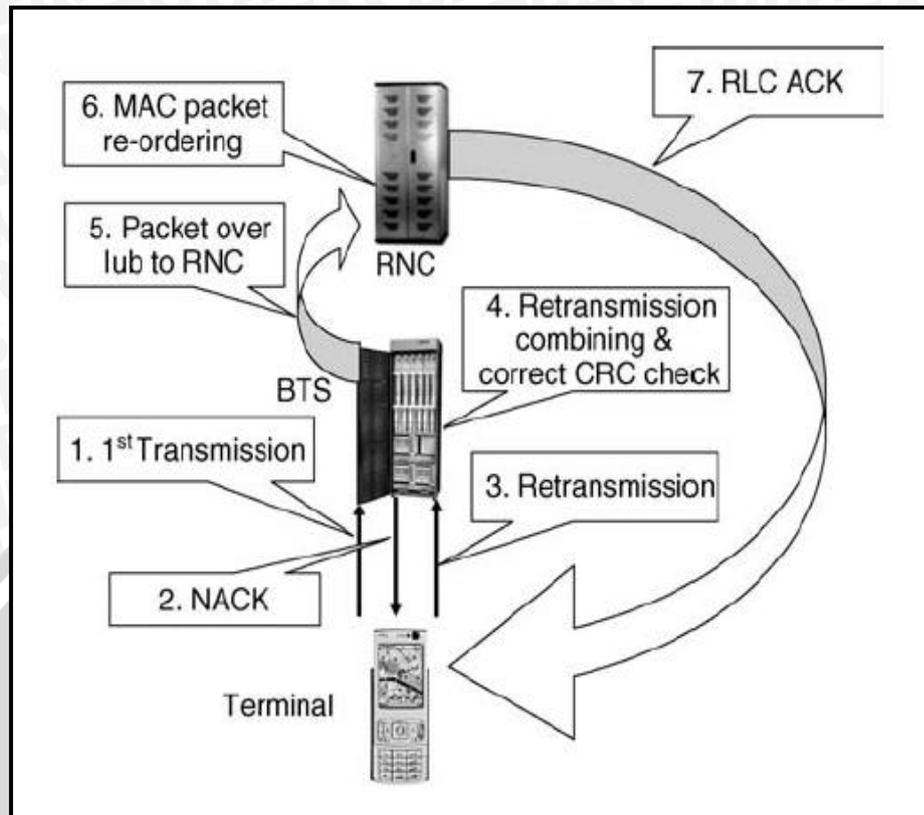
HARQ (*Hybrid Automatic Repeat Request*) merupakan penggabungan dua teknik *error correction* yaitu ARQ (*Automatic Repeat Request*) yang memiliki keandalan sistem dan FEC (*Forward Error Correction*) yang mampu mempertahankan *throughput* dalam keadaan konstan pada kondisi kanal yang memburuk. HARQ mengadopsi protokol *stop and wait*, yaitu paket data hanya dapat ditransmisikan setelah paket data sebelumnya (baik mengalami retransmisi atau tidak) sukses ditransmisikan. Gambar 2.3 memperlihatkan proses *stop and wait*.



Gambar 2.3 Proses *stop and wait*

Sumber: I Putu (3)





**Gambar 2.4** Kontrol retransmisi pada sistem *Release '99* dan *Enhanced Uplink*

**Sumber:** Harri Holma and Antti Toskala (2007:406)

### 2.2.1.1 HARQ I

Pada HARQ I, masing-masing paket dikodekan untuk *error detection* (CRC/*Cyclic Redundancy Check*) dan *error correction* (FEC). Jika terdapat *error* setelah FEC *decoding*, *receiver* membuang paket yang salah kemudian meminta *transmitter* untuk mengirim ulang seluruh data yang diterima sebelumnya. Proses *retransmisi* terus dilakukan hingga data yang dikirim benar. Kelemahan HARQ I adalah adanya *Overhead* yang selalu menyertai dalam setiap transmisi. *Overhead* tersebut merupakan *extra parity check bits* untuk *error correction*. Namun bila terdapat *noise* pada kanal tersebut, penggunaan *parity check* akan sia-sia.

### 2.2.1.2 HARQ II

HARQ II merupakan penyempurnaan HARQ I meski masing-masing paket tetap dikodekan untuk *error detection* dan *correction* yang sama pada HARQ I. *Receiver* menyimpan paket yang salah pada buffer agar dapat digunakan kembali hingga proses transmisi selesai.

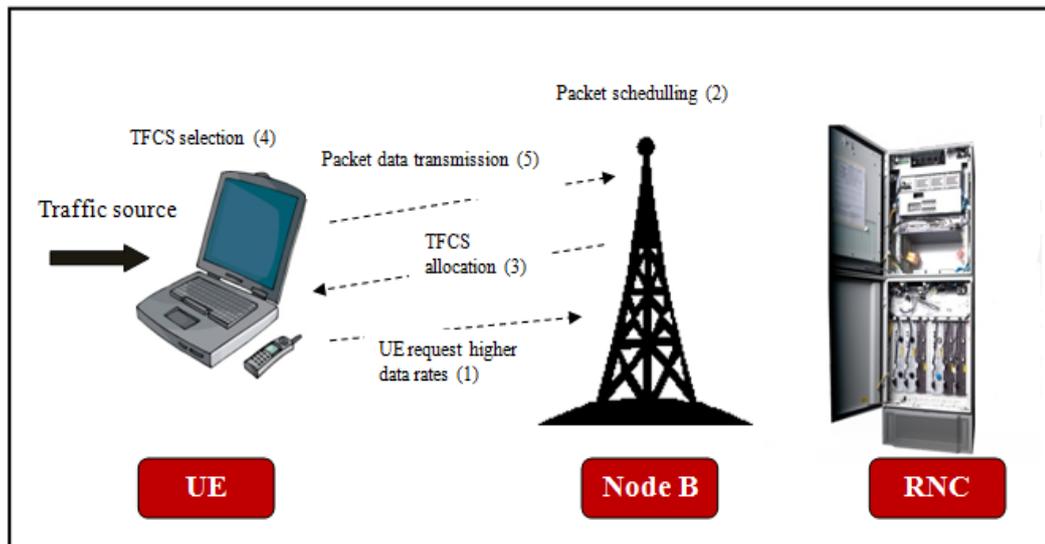
HARQ II mempunyai IR (*Incremental Redundancy*) dan *chase combining* sebagai pelengkap.

Pada proses transmisi, IR akan mengatur agar hanya sebagian *parity bit* saja yang diretransmisi. *Parity bit* akan di retransmisi hingga *receiver* menerima *parity bit* yang lengkap. Pada IR, paket hasil retransmisi membawa *parity bit* untuk *error correction*.

Prinsip *chase combining* adalah adanya pengiriman sejumlah paket data yang terkodekan dan kemampuan *decoder* untuk mengkombinasikan beberapa paket yang sama dari proses retransmisi. Setelah menerima data, *receiver* memeriksa paket data pada bagian setelah proses FEC *decoding*. Jika terdapat *error*, *nack* akan dikirim ke *transmitter* agar terjadi retransmisi. *Soft decision* data *sequence* disimpan pada *buffer receiver* untuk kemudian dikombinasikan dengan paket retransmisi secara *frame per frame*.

### 2.2.2 Node B Controlled Scheduling

*Node B controlled scheduling* digunakan untuk mengurangi *delay* pada WCDMA dengan memindahkan *uplink scheduling* dan *data rate* pada *node B* sesuai batasan RNC (HSUPA). Seperti pada gambar 2.5, UE meminta giliran *packet scheduling* pada *Node B* melalui *uplink signaling* (1) kemudian *packet scheduler* pada *Node B* menentukan TFC yang akan dialokasikan untuk UE (2). *Node B* kemudian akan mengalokasikan TFC untuk UE (3) sedangkan UE akan memilih TFC yang sesuai dengan daya yang memiliki (4). Selanjutnya UE akan mengirimkan datanya (5).



Gambar 2.5 Packet scheduling pada Node B

Sumber: Camilio OM (2006:26)

### 2.2.3 TTI (*Transmission Time Interval*)

Pada jaringan berbasis paket data, informasi yang akan ditransmisikan dibentuk terlebih dahulu menjadi paket-paket data yang ada pada jaringan WCDMA dinamakan TTI (*Transmission Time Interval*) dengan ukuran 10 ms, 20 ms atau 40 ms berdasarkan 3GPP *released 99*. HSUPA menyediakan TTI 10 ms untuk semua UE dan 20 ms untuk UE tertentu berdasarkan kategori yang digunakan. TTI yang lebih pendek diharapkan mampu mengurangi *delay* transmisi data, mengurangi besarnya waktu tunggu transmisi dan selanjutnya, mengurangi waktu proses dan mengurangi waktu *roundtrip* pada Node B yang menggunakan HARQ.

### 2.2.4 Jenis-Jenis Kanal pada HSUPA

#### 2.2.4.1 Kanal Transport

Kanal *transport* pada HSUPA dinamakan E-DHC (*Enhanced Dedicated Channel*). E-DHC memiliki karakteristik digunakan pada proses HARQ, hanya memiliki satu *transport block* tiap TTI, memiliki satu E-DHC dan CCTrCH (*Coded Composite Transport Channel*) tiap UE, lapisan MAC dapat memultipleks beberapa layanan yang berbeda ke dalam sebuah E-DCH, E-DCH dan DCH bias

terdapat pada sebuah UE dengan syarat kecepatan DHC tidak lebih dari 64 kbps dan menggunakan CCTrCH terpisah.

#### 2.2.4.2 Kanal Fisik

HSUPA memiliki beberapa kanal fisik sebagai bagian dari kanal *uplink* dan *downlink*, yaitu:

##### 1. Kanal Uplink

- a. E-DPDCH (*Enhanced Dedicated Physical Data Channel*) yang membawa *payload* dan memungkinkan terdapatnya *permintaan scheduling* dari UE ke *Node B*.
- b. E-DPCCH (*Enhanced Dedicated Physical Control Channel*) yang membawa informasi kontrol yang dibutuhkan untuk *decode payload* yang dibawa E-DPDCH dan membawa indikasi dari UE ke *Node B* saat sumber memenuhi syarat.

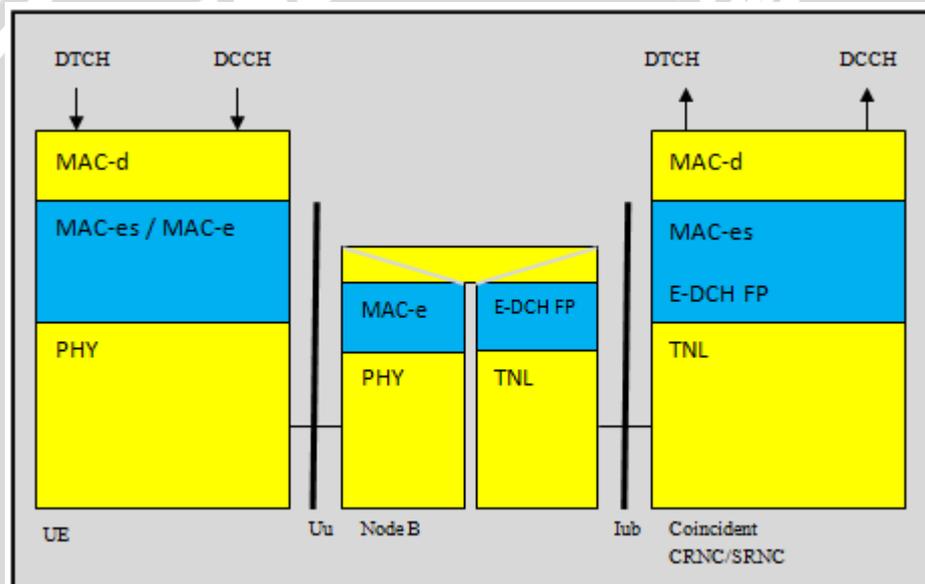
##### 2. Kanal Downlink

- a. E-HICH (*Enhanced HARQ Acknowledgement Indicator Channel*) yang memberi umpan balik pada UE tentang transmisi data sebelumnya, membawa Ack dan Nack.
- b. E-RGCH (*Enhanced Relative Grant Channel*) yang membawa perintah sederhana untuk *increase* (up), *decrease* (down) atau mempertahankan (hold) *grant*.
- c. E-AGCH (*Enhanced Absolute Grant Channel*) yang membawa maksimum E-DPCCH/DPDCH dan membawa informasi pengontrol HARQ.

#### 2.2.5 Protokol HSUPA

Protokol baru pada HSUPA meliputi MAC-es/MAC-e pada UE, MAC-e pada *Node B* dan MAC-es pada RNC. MAC-e pada UE dan *Node B* bertanggung jawab atas proses HARQ dan *scheduling*, *multiplexing* (MAC-e UE) dan *demultiplexing* (MAC-e *Node B*). MAC-es pada UE dan RNC digunakan untuk proses *in-sequence delivery*, khususnya saat terjadi pentransmisian paket-paket data melalui *Node B* yang berbeda (saat terjadi *handover*).

Pada UE, MAC-d bertanggung jawab untuk memetakan 1/lebih *dedicated logical channel* (DDTCH dan DDCH) dari RLC ke seluruh *dedicated transport channel* (E-DCH). MAC-es akan mengikutsertakan 6 bit TSN (*Transmit Sequence Number*) ke dalam PDU (*Protocol Data Unit*) dari MAC-es dengan 6 bit DDI (*Data Description Indicator*) yang mengidentifikasi setiap kanal transport. Kemudian MAC-e melakukan *scheduling* pada paket yang ditransmisikan berdasarkan TFC (dan juga daya yang dimilikinya), dan mentransmisikan berdasarkan informasi *absolute grant*. Hasil proses MAC-e dipetakan ke dalam kanal fisik E-DPDCH untuk ditransmisikan dan setelahnya disimpan dalam HARQ *handler* guna proses retransmisi (jika dibutuhkan).



**Gambar 2.6** Arsitektur protokol HSUPA

**Sumber:** Rohde Schwarz

Untuk setiap UE berkonfigurasi E-DCH, hanya terdapat satu MAC-e pada *Node B* dan MAC-es pada SRNC. Pada *Node B*, frame yang datang akan dikodekan oleh MAC-e. HARQ *handler* akan mengirimkan ACK (paket sukses) dan NACK (paket gagal) melalui E-HICH. MAC-e akan *men-demultiplex* frame ke dalam kanal transport berdasarkan DDI pada *header*-nya. Pada RNC, MAC-es akan menerima setiap paket pada kanal transport dan melakukan *reordering* berdasarkan TSN-nya. MAC-d akan memisahkan beberapa kanal logika yang berbeda (Camillo OM, 2006:19-21).

### 2.2.6 Terminal HSUPA

Terminal *user* yang dipakai pada HSUPA dinamakan *User Equipment* (UE). Masing-masing terminal tersebut dibedakan berdasarkan kategori yang digunakan. Terdapat 6 kategori pada HSUPA (tabel 2.2)

**Tabel 2.2** Data *rate* HSUPA kategori

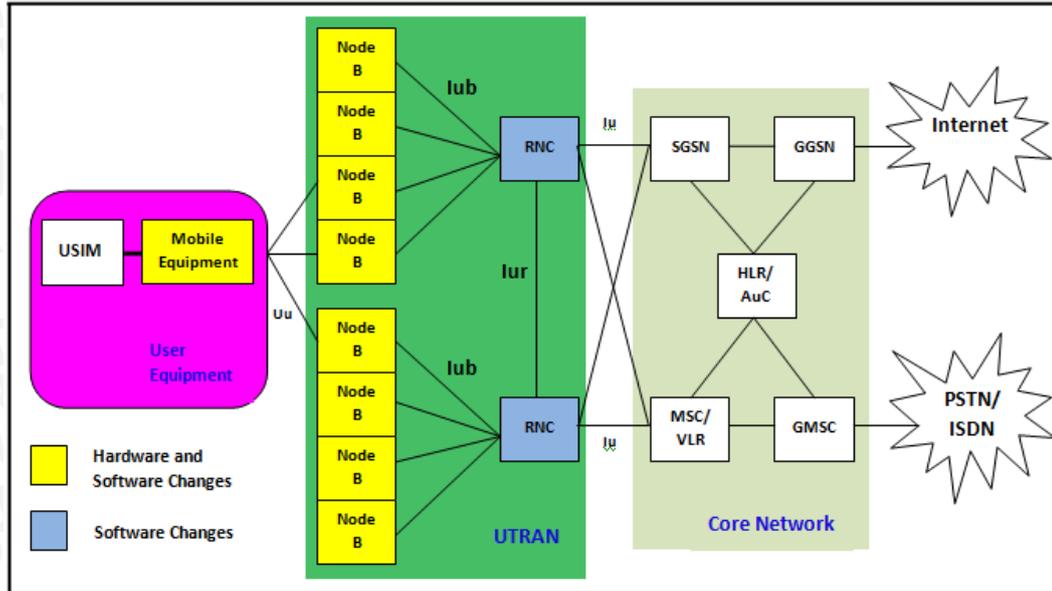
Category	Maximum number of E-DPDCH and smallest spreading factor	Supported TTI	Maximum data rate with a 10 ms TTI	Maximum data rate with a 2 ms TTI
1	1xSF4	10 ms	0,71 Mbps	N/A
2	2xSF4	2 and 10 ms	1,45 Mbps	1,45 Mbps
3	2xSF4	10 ms	1,45 Mbps	N/A
4	2xSF2	2 and 10 ms	2 Mbps	2,91 Mbps
5	2xSF2	10 ms	2 Mbps	N/A
6	2xSF2 + 2xSF4	2 and 10 ms	2 Mbps	5,76 Mbps

**Sumber:** Pedro Baptista (2007:14)

### 2.3 Arsitektur Jaringan HSUPA

Jaringan HSUPA terdiri dari beberapa bagian antara lain *mobile equipment*, *Node B* dan RNC yang tersambung pada jaringan pusat (*Core Network*). Perbedaan WCDMA-HSUPA dengan WCDMA adalah pada *uplink scheduling* dan *rate control* yang terletak pada *Node B* sesuai dengan RNC.

*Mobile Equipment* (atau dapat juga disebut dengan UE) merupakan *handset user* yang berfungsi sebagai pengirim maupun penerima panggilan, sedangkan *Node B* merupakan istilah yang digunakan pada jaringan WCDMA untuk menandai BTS (*Base Transceiver Station*). *Node B* menangani sambungan antar *handset* pengguna atau disebut sebagai *interface* antar *handset user* kemudian *Node B* akan dikontrol langsung oleh RNC. Jaringan WCDMA-HSUPA memakai konsep HARQ, TTI dan *fast scheduling* pada *Node B*. TTI 2 ms dan 10 ms pada HSUPA relatif lebih pendek dari WCDMA (10 ms, 20 ms atau 40 ms). TTI yang relatif lebih pendek ini akan mengurangi *delay* transmisi data, mengurangi besarnya waktu proses dan juga mengurangi waktu *roundtrip* pada *Node B* yang menggunakan HARQ.

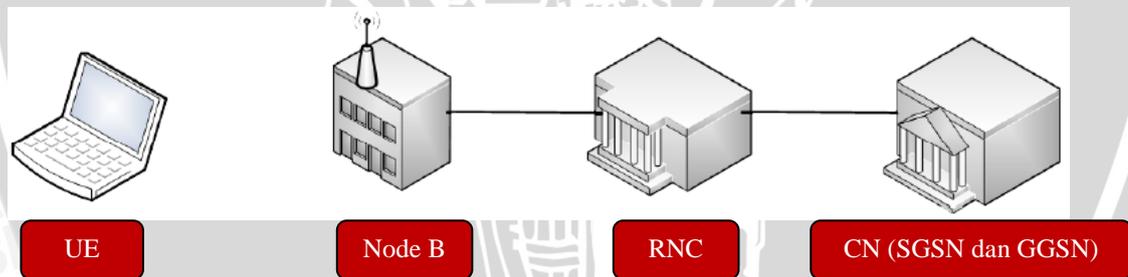


Gambar 2.7 Arsitektur Jaringan HSUPA – WCDMA (UMTS)

Sumber: Syed Ismail (84)

Penerapan HSUPA (dimungkinkan *bit rate* yang lebih tinggi) pada jaringan WCDMA akan berpengaruh juga pada area cakupannya. Model jaringan fokus pada arsitektur UE-UTRAN-CN.

### 2.3.1 Definisi Umum UTRAN



Gambar 2.8 UE-UTRAN-CN architecture

Sumber: Harri Holma

1. Berstandart pada jaringan WCDMA,
2. Mendukung UE, Node B, RNC, GGSN dan SGSN,
3. Mendukung scenario hardhandover, softhandover, dan softer handover,
4. UTRAN mengandung beberapa element set dari Radio Network Subsystem (RNS) yang terhubung dengan Core Network melalui interface Iu,

5. RNS mengandung beberapa element set Radio Network Controller (RNC) didalam nya lagi terdapat beberapa Node B yang terhubung dengan RNC melalui Iub interface,
6. Sebuah Node B bisa mendukung mode FDD, mode TDD atau dual mode operation,
7. RNC bertanggung jawab untuk memutuskan proses Handover dan hal ini membutuhkan signaling dari UE,
8. RNC membandingkan sebuah proses combining/splitting untuk mendukung macro diversertiy di antara Node B yang berbeda, dan
9. Pada UTRAN ,RNC dapat dihubungkan dengan interface logical Iur dan Iur secara bersamaan.

#### 2.4 Video Conference

*Video conference* adalah salah satu aplikasi *Multimedia* yang memungkinkan komunikasi data, suara, dan gambar yang bersifat *duplex* serta *real time*. Seperti namanya, bentuk dari aplikasi ini adalah percakapan *via video* dan *audio* antar pengguna secara langsung dan diharapkan dapat menggantikan fungsi tatap muka secara langsung.

Ada dua tipe *video conference*, *point-to-point* dan *multipoint*. Dimana *video conference point-to-point* merupakan metode sederhana yang menggunakan dua buah komputer untuk saling terhubung menggunakan *single IP address*. Biasanya *point-to-point* digunakan untuk *video phone calls* yang sifatnya lebih personal. *Video conference point-to-point* menggunakan *web cam* dan *microphone* yang terpasang di masing-masing komputer pengguna sehingga pengguna bisa berbicara dan saling melihat lawan bicara.

##### 2.4.1 Proses Video Conference

*Video conference* bekerja menggunakan berbagai gabungan dari teknologi-teknologi yang berbeda. Beberapa teknologi diantaranya merupakan perangkat keras (*hardware*) dan teknologi lainnya berupa perangkat lunak (*software*). Berikut merupakan teknologi beserta perangkat yang digunakan untuk membangun sebuah *video conference*.

#### 2.4.1.1 Video Input

*Video input* biasanya terdiri dari perangkat seperti *web cam* atau kamera video. Untuk *video conference* yang besar biasanya terdapat dua atau lebih perangkat untuk *video input*.



Gambar 2.9 Webcam

#### 2.4.1.2 Video Output

*Video output* bisa berupa TV set standar, set layar digital, ataupun menggunakan perangkat proyektor digital.

#### 2.4.1.3 Audio Input

Untuk *audio input* dalam *video conference* tentunya digunakan perangkat *microphone* atau *headset*. *Microphone* atau *headset* dapat dipasang di masing-masing peserta *conference* atau dapat dipasang secara terpusat.



Gambar 2.10 Headset

#### 2.4.1.4 Audio Output

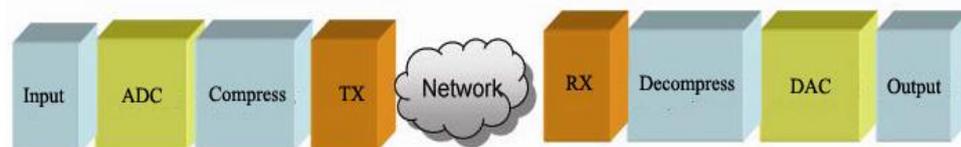
*Audio Output* biasanya terdiri dari *speakers*. *Speakers* bisa dalam bentuk *headphone* yang terpasang di masing-masing peserta ataupun dipasang secara terpusat.



Gambar 2.11 Speaker

#### 2.4.1.5 Codec

*Codec* berguna untuk melakukan konversi data analog ke digital atau digital ke analog. ADC (*analog-to-digital converter*) sebuah kamera video mengubah sinyal analog menjadi sinyal digital, yang kemudian sinyal tersebut melewati sebuah kompresor video untuk ditransmisikan secara digital. Sebuah perangkat penerima kemudian menjalankan sinyal tersebut melalui dekompresor video, kemudian sinyal tersebut dikonversikan ke bentuk awal menggunakan DAC (*digital-to-analog converter*).



Gambar 2.12 Codec

Sumber: WordPress.com

#### 2.4.1.6 Perangkat Lunak

Perangkat lunak berupa aplikasi yang digunakan untuk menampilkan video dan audio di dalam *video conference*, contohnya seperti aplikasi Yahoo Messenger, MSN Messenger, Skype dan lain-lainnya.

*Video conference* di transmisikan melalui koneksi internet biasa atau melalui koneksi broadband yang disediakan oleh ISP (*Internet Service Provider*). Disini penggunaan perangkat lunak yang berfungsi mengkompresi data juga diperlukan agar tidak terjadi koneksi yang lamban akibat penggunaan sejumlah besar *bandwidth* yang sedang digunakan oleh data video dan audio melalui jaringan.



Gambar 2.13 Skype

## 2.5 Format Gambar

Informasi gambar terdiri dari tiga elemen yaitu ukuran gambar dan jumlah garis yang digunakan untuk membangkitkan gambar, kecepatan pengulangan pada gambar, dan jumlah resolusi yang diperlukan untuk gambar. Format gambar yang digunakan pada aplikasi *video conference* antara lain CIF (*Common Intermediate Format*) dan QCIF (*Quarter Common Intermediate Format*).

Tabel 2.3 Konsumsi *bandwidth* aplikasi *video conference*

Nominal Bandwidth	Size	Frame rate	Actual Bandwidth
128 Kbps	QCIF	5 ms	140 Kbps
174 Kbps	CIF	8 – 10 ms	189 Kbps
384 Kbps	CIF	15 ms	408 Kbps
768 Kbps	CIF	30 ms	810 Kbps

Sumber: Norcahyanto (2008:18)

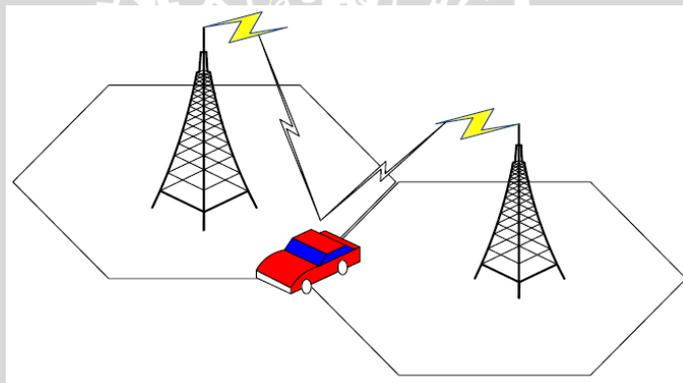
CIF memiliki kualitas sama dengan VCR (*Video Cassete Recorder*). Format ini direkomendasikan untuk aplikasi *multi video conference* karena

memiliki *bandwidth* yang cukup besar sekitar 384 Kbps. QCIF merupakan format alternatif untuk aplikasi *video*, dimana resolusinya adalah setengah dari resolusi QCIF sehingga kualitasnya dibawah CIF. Sehingga *bandwidth* dengan 15 *frame* per detik maupun 10 *frame* per detik masih dianggap memenuhi syarat untuk aplikasi *video conference*.

## 2.6 Handover

*Handover* merupakan proses transfer suatu *ongoing call* atau data session dari suatu kanal yang terhubung dalam satu inti jaringan ke kanal lain. Pada komunikasi satelit, istilah tersebut diartikan pengalihan tanggung jawab kontrol satelit dari satu stasiun bumi ke stasiun yang lain tanpa kesalahan (*loss*) atau interupsi layanan yang berstandar 3GPP yang berasal dari teknologi Eropa seperti UMTS.

### 2.6.1 Soft Handover



Gambar 2.14 Soft Handover

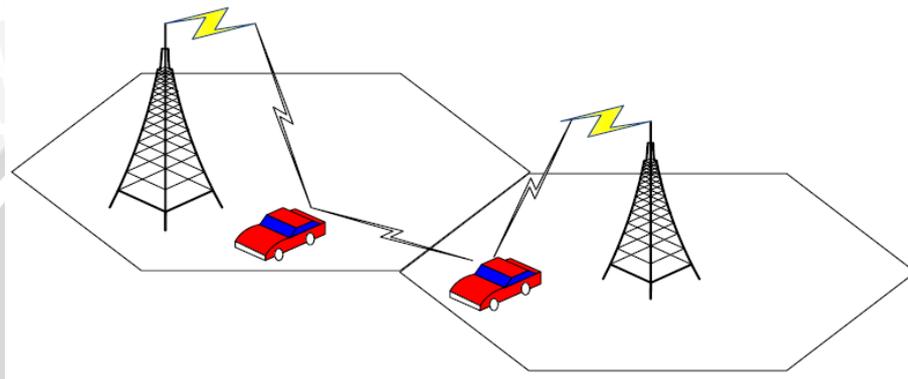
Sumber: Zeng and Agrawal

*Soft handoff* adalah suatu metode dimana kanal pada sel sumber tetap tersambung dengan user sementara secara paralel juga menghubungi kanal pada sel target. Pada kasus ini, sambungan ke target harus berhasil dahulu sebelum memutuskan sambungan dengan sel sumber, karena itulah *soft handoff* juga disebut “*make-before-break*”. Interval selama terjadinya dua sambungan dilakukan secara paralel bisa saja singkat maupun substansial (tergantung kondisi yang memungkinkan).

Ketika panggilan dalam keadaan *soft handoff*, sinyal yang terbaik dari semua penggunaan kanal dapat dimanfaatkan untuk panggilan pada saat itu atau

semua sinyal dikombinasikan agar dapat menghasilkan duplikat sinyal yang lebih baik. Kemudian yang lebih menguntungkan adalah, ketika kedua performa dikombinasikan pada *downlink (forward link)* dan *uplink (reverse link)* maka *handoff* tersebut menjadi lebih halus (*softer*). *Softer handoff* dapat dilakukan apabila sel yang mengalami *handoff* berada dalam satu situs sel.

### 2.6.2 Hard Handover



Gambar 2.15 Hard Handover

Sumber: Zeng and Agrawal

*Hard handoff* adalah suatu metode dimana kanal pada sel sumber dilepaskan dan setelah itu baru menyambung dengan sel tujuan. Sehingga koneksi dengan sel sumber terputus sebelum menyambung dengan sel target untuk alasan tersebut *hard handoff* juga dikenal dengan sebutan "*break-before-make*". *Hard handoff* dimaksudkan untuk meminimalkan gangguan panggilan secara instan. Suatu *hard handoff* dilakukan oleh jaringan selama panggilan berlangsung.

## 2.7 Parameter Kinerja Jaringan HSUPA Aplikasi Video Conference

Parameter kinerja yang akan dianalisis pada skripsi ini adalah sebagai berikut:

### 2.7.1 Delay

*End to end delay* adalah waktu yang diperlukan oleh suatu paket data yang berasal dari *source node* hingga mencapai *destination node*. *End to end delay* secara tidak langsung berhubungan dengan kecepatan transfer data suatu jaringan. Beberapa kontribusi jaringan dalam *end to end delay* adalah :

1. *Propagation delay* adalah waktu yang diperlukan oleh suatu info untuk melintasi suatu line dengan jarak tertentu. *Propagation delay* ditentukan oleh jarak dan kecepatan cahaya.
2. *Transmission delay* adalah waktu yang diperlukan oleh sebuah paket data untuk melintasi suatu media. *Transmission delay* dipengaruhi oleh kecepatan media dan besar paket data.
3. *Processing delay* adalah waktu yang diperlukan sebuah network untuk memproses data real menjadi bitbit data yang akan dikirimkan.
4. *Queing delay* adalah waktu yang diperlukan oleh sebuah paket dalam suatu antrian.
5. *Round trip delay* adalah waktu yang diperlukan oleh suatu paket data dari *source node* ke *destination node* hingga kembali lagi ke *source node*. *Round trip delay* dihasilkan dari proses pengetesan keterjangkauan (*reachable*) suatu *node* yang berada dilokasi yang berbeda. Dengan asumsi waktu yang diperlukan dari *source node* ke *destination node* sama dengan waktu yang diperlukan dari *destination node* ke *source node*.

### 2.7.2 Throughput

*Throughput* adalah jumlah rata-rata paket yang sukses diterima atau dikirimkan oleh saluran penerima atau pemancar per detik. *Throughput* merupakan salah satu parameter yang menunjukkan kinerja dari suatu sistem komunikasi data.

### 2.7.3 SNR (Eb/No)

SNR (*Signal to Noise Ratio*) merupakan Perbandingan (*ratio*) antara kekuatan Sinyal (*signal strength*) dengan kekuatan Derau (*noise level*). Nilai SNR dipakai untuk menunjukkan kualitas jalur (medium) koneksi. Makin besar nilai SNR, makin tinggi kualitas jalur tersebut. Artinya, makin besar pula kemungkinan jalur itu dipakai untuk lalu-lintas komunikasi data & sinyal dalam kecepatan tinggi. Nilai SNR suatu jalur dapat dikatakan pada umumnya tetap, berapapun kecepatan data yang melalui jalur tersebut. Satuan ukuran SNR adalah *decibel*

(dB). Nilai dianggap bagus jika bernilai  $\geq 7$  dB dan nilai Loop Attenuation  $\leq 18$  dB.

**Tabel 2.4** Konsumsi SNR aplikasi *video conference*

No	Nilai (dB)	Keterangan
1.	29 – ke atas	<i>Outstanding</i> (bagus sekali).
2.	20 – 28,9	<i>Excellent</i> (bagus). Koneksi stabil.
3.	11 – 19,9	<i>Good</i> (baik). Sinkronisasi sinyal ADSL dapat berlangsung lancar.
4.	7 – 10,9	<i>Fair</i> (cukup). Rentan terhadap variasi perubahan kondisi pada jaringan.
5.	0 – 6,9	<i>Bad</i> (buruk). Sinkronisasi sinyal gagal atau tidak lancar (terputus-putus).

Sumber: Azilmi

## 2.8 Topologi Jaringan

Topologi adalah suatu aturan bagaimana cara menghubungkan komputer satu sama lain secara fisik dan pola hubungan antara komponen-komponen yang berkomunikasi melalui media/peralatan jaringan yang meliputi *server*, *workstation*, *hub/switch* dan pengkabelannya. Topologi jaringan adalah susunan atau pemetaan interkoneksi antara *node*, dari suatu jaringan, baik secara fisik (riil) dan logis (virtual). (Dede Sopandi,2008)

### 2.8.1 Point to Point

*Point to point* memiliki konektivitas paling sederhana dimana konektivitas bekerja antar dua titik akhir pada sebuah jaringan. Begitu sederhananya jaringan ini, sehingga seringkali tidak dianggap sebagai suatu jaringan tetapi hanya merupakan jalur komunikasi biasa.

Kelebihan topologi *point to point*:

1. Mudah menghubungkan antar komputer
2. Membutuhkan kabel yang pendek.

Kekurangan topologi *point to point*:

1. Seluruh jaringan akan mati bila kabel utama terputus.
2. Sulit mencari dan memperbaiki kerusakan apabila terjadi kerusakan pada jaringan.

3. Tidak mungkin dimplementasikan pada jaringan dengan banyak komputer.

Gambar 2.16 merupakan contoh jaringan yang menerapkan topologi *point to point*:

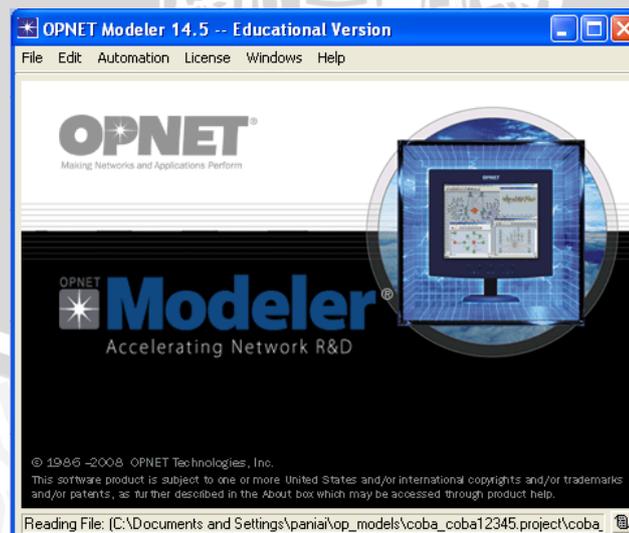


**Gambar 2.16** Topologi *point to point*

(Sumber: <http://www.inetdaemon.com/tutorials/networking/lan/topology.shtml#pointtopoint>)

## 2.9 Optimized Network Engineering Tool (OPNET) Modeler v.14.5

OPNET Modeler v.14.5 adalah sebuah *network simulator* yang dirancang oleh OPNET Technologies Inc. OPNET Modeler mengakselerasikan R&D *network*, mengurangi *time-to-market*, dan meningkatkan kualitas produk. Dengan menggunakan simulasi, *network designers* dapat mengurangi biaya penelitian dan memastikan kualitas produk yang optimal. Teknologi terbaru OPNET Modeler menyediakan sebuah lingkungan untuk mendesain protokol dan teknologi juga menguji dan mendemonstrasikan dengan skenario yang realistis sebelum diproduksi. OPNET Modeler digunakan perusahaan perlengkapan jaringan terbesar di dunia untuk meningkatkan desain dari *network devices*, teknologi seperti VoIP, TCP, OSPFv3, MPLS, IPv6 dan lain-lainya.



**Gambar 2.17** Tampilan awal OPNET Modeler 14.5 – *Educational Version*

(Sumber : <http://www.opnet.com>)

Di dalam simulasi jaringan berbasis IP khususnya IPv6 dengan mempergunakan *simulator* OPNET, hal-hal yang perlu dilakukan antara lain:

a. Konfigurasi Jaringan

Di dalam *simulator* OPNET harus dilakukan penggambaran model jaringan yang akan disimulasikan. Konfigurasi yang digambarkan disesuaikan dengan model jaringan yang akan disimulasikan. Secara umum untuk menggambarkan suatu jaringan berbasis IP antara lain terdapat: *router*, *bridge/switch*, *hub*, LAN, *link* baik yang dipergunakan untuk menghubungkan antar *router* ataupun hubungan ke *user*, *workstation*, *application server*, dan lain-lain. Kelengkapan suatu model akan tergantung kepada kebutuhan dan kerumitan yang diinginkan.

b. *Profile User*

Dipergunakan untuk menggambarkan *profile* dari *user* yang disimulasikan di dalam model tersebut. Sebagai contoh *profile* karyawan akan memiliki *profile* sesuai dengan kondisi karyawan di suatu perusahaan apakah dia memiliki aksesibilitas untuk menjalankan semua aplikasi di dalam jaringan perusahaan tersebut atau terbatas.

c. Layanan

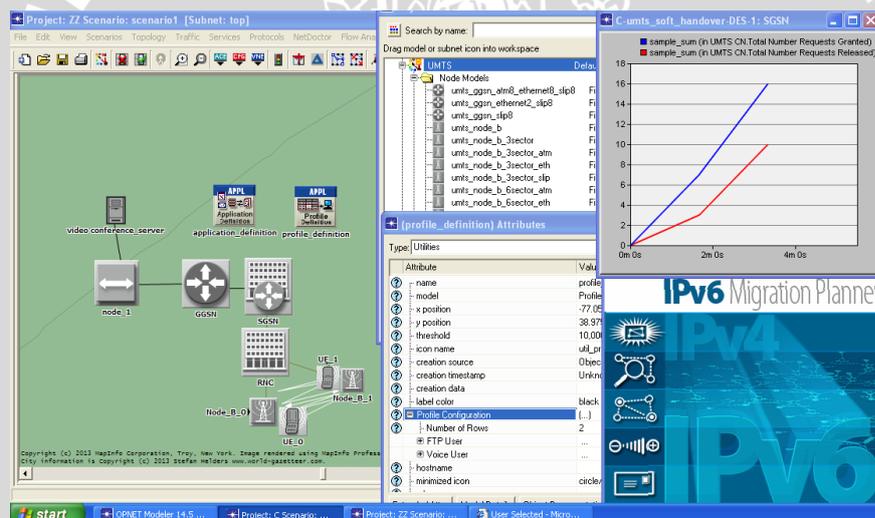
Dipergunakan untuk menggambarkan aplikasi/layanan apa saja yang dijalankan di dalam jaringan tersebut. Di dalam simulator OPNET, aplikasi yang dapat dijalankan antara lain *Email*, TELNET, *Database*, FTP, Print, VoIP, *remote login*, *video conference* ataupun aplikasi lain yang disesuaikan dengan kebutuhan pelanggan. Aplikasi tersebut merupakan aplikasi yang default telah disediakan oleh OPNET, dan masing-masing terdiri atas aplikasi yang dijalankan secara umum, rendah, berat ataupun dapat di *setting* sesuai kebutuhan.

### 2.9.1 Kelebihan OPNET Modeler v.14.5

Berikut adalah beberapa alasan digungkannya OPNET Modeler versi 14.5 dalam penelitian ini (Jinhua Guo, 2007):

- OPNET menyediakan *Graphic User Interface* (GUI) yang sangat nyaman dan mudah untuk dipelajari,

- OPNET dapat digunakan untuk membuat model pada seluruh jaringan, termasuk router, switch, protokol, server, dan aplikasi lain yang mendukung. Berbagai macam system komunikasi mulai LAN tunggal hingga antar jaringan global dapat didukung,
- *Software* OPNET (dengan model *source code*) tersedia secara gratis untuk penelitian akademis dan mengajar masyarakat,
- *Discrete event simulation (DES) engine* pada OPNET untuk simulasi jaringan adalah tercepat. Biasanya diperlukan waktu beberapa menit untuk menyelesaikan simulasi percobaan laboratorium,
- OPNET memiliki komunitas pengguna yang besar. OPNET telah digunakan lebih dari 500 perusahaan, penyedia layanan, dan organisasi pemerintah di seluruh dunia. siswa yang memiliki pengalaman dengan simulator OPNET akan memiliki peluang kerja jauh lebih baik di masa depan dalam industri.



**Gambar 2.18** Tampilan proses simulasi dengan menggunakan OPNET Modeler  
(Sumber :<http://www.opnet.com>)

### 2.9.2 Discrete event simulation (DES)

Simulasi adalah proses mendesain suatu model yang dikomputerisasi dari sebuah sistem atau sebuah proses dan menghubungkan percobaan dengan model tersebut untuk tujuan memahami perilaku dari system atau mengevaluasi strategi yang beragam dari operasi pada system tersebut.

Kelebihan dari proses simulasi adalah :

- dapat mengendalikan percobaan, dengan mengubah parameter tertentu tanpa harus mengubah atau mengganggu system yang telah ada,
- dapat lebih menghemat waktu,
- merupakan perangkat yang efektif digunakan dalam program pelatihan

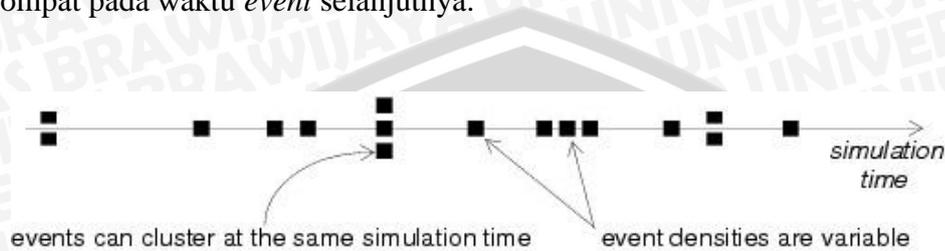
*Discrete event simulation* (DES) merupakan suatu model simulasi dimana perubahan statusnya terjadi pada titik-titik diskrit dalam waktu yang dipicu oleh suatu kejadian (*event*). Dalam titik waktu tersebut akan terjadi suatu *event*, dimana *event* didefinisikan sebagai suatu kejadian yang dapat mengubah kondisi suatu sistem. Contohnya operator yang sebelumnya menganggur (*idle*), setelah kedatangan suatu pelanggan akan menjadi sibuk. Atau sebaliknya dari kondisi sibuk ke kondisi *idle*.

Berikut adalah beberapa kejadian yang dapat dimodelkan sebagai sebuah *event*:

- menerima sebuah paket atau sebuah perintah dalam suatu proses,
- mulai atau berakhirnya penransmisian atau penerimaan paket-paket pada suatu *link*,
- pembangkitan sebuah pesan baru, perintah, atau tugas oleh sebuah proses aplikasi,
- kegagalan (atau *recovery* dari kegagalan) pada suatu perangkat.

Ada dua pendekatan pokok yang digunakan untuk meningkatkan waktu simulasi yaitu mempercepat waktu *event* berikutnya dan mempercepat interval waktu tetap. Dengan pendekatan mempercepat *event* berikutnya, waktu simulasi diinisialisasi dengan (nol) 0 kemudian banyaknya kejadian dari suatu *event* di masa depan ditentukan. Waktu simulasi kemudian dipercepat ke waktu kejadian *event* paling dekat di masa depan, kemudian titik-titik kondisi sistem diperbarui ke setiap kejadian yang telah terjadi sehingga banyaknya *event* di masa depan juga diperbarui. Kemudian waktu simulasi dipercepat, kondisi sistem diperbarui, banyaknya *event* masa depan ditentukan dst. Proses dari waktu simulasi yang dipercepat ini dari waktu *event* satu ke yang lain kontinu sampai beberapa kondisi berhenti dan telah terpenuhi. Sejak semua perubahan kondisi hanya terjadi pada *discrete event simulation*, dimana periode tidak aktif dilompati dari satu

waktu *event* ke waktu *event* lain. Waktu simulasi hanya berubah ketika *event* baru terjadi, selama satu *event* terjadi, waktu tidak dapat berubah. Waktu simulasi sekarang didefinisikan sebagai waktu *event* saat ini dieksekusi. Waktu simulasi tidak perlu dipercepat dengan besar yang sama pada setiap *event*, namun melompat pada waktu *event* selanjutnya.



**Gambar 2.19** Distribusi event pada *timeline* simulasi  
(Sumber : OPNET tutorial.pdf)

OPNET Modeler mendukung pemodelan dalam sistem terdistribusi, dan OPNET juga memperbolehkan dilakukannya *multiple-event* terjadi secara simultan dalam simulasi dan mempengaruhi komponen-komponen yang berbeda dari sistem. Oleh karena itu, dimungkinkan untuk terjadinya beberapa *event* berbeda dalam waktu simulasi yang sama.

