

BAB II

TINJAUAN PUSTAKA

2.1 Umum

Layanan *video on demand* (VOD) merupakan salah satu jenis kegiatan *video streaming*. Parameter video yang meliputi format video, *frame per second*, *bitrate* dan resolusi video menentukan kinerja VOD. Kinerja VOD juga ditentukan melalui protokol, perangkat serta jaringan yang digunakan. *High Speed Downlink Packet Access* (HSDPA) merupakan jaringan UMTS dengan kecepatan *downlink* hingga 13,6 Mbps dan *bandwidth* 5 MHz. Kinerja VOD yang dilewatkan pada jaringan HSDPA dapat diukur melalui QoS, yakni *delay end to end*, *packet loss* dan *throughput*. Perbandingan nilai QoS yang didapat dari hasil perhitungan secara teoretis dan hasil pengamatan digunakan untuk analisis. Analisis yang didapatkan dibandingkan dengan rekomendasi dari ITU terkait QoS yang digunakan.

2.2 Video

Video adalah informasi yang berisi gambar dan suara serta memiliki ciri khas gambar bergerak dengan kecepatan tertentu (*frame per second*) (jiscdigitalmedia.ac.uk, 2014)

2.2.1 Video Format

Video *format* mencakup dua hal, yakni *Containers* (disebut sebagai *Wrappers*) dan *Codecs* (singkatan *codec/decoder*). *Codec* digunakan dalam *container*. *Container* mendeskripsikan struktur dari *file* dan *codec* menentukan aturan yang digunakan. *Container* mengemas video dan komponennya (audio/metadata) menjadi *file* dengan ekstensi, seperti .AVI, .MP4 atau .MOV.

Tabel 2.1 Daftar *Container*

| Jenis Container | Keterangan |
|---------------------------------------|---|
| AVI (<i>Audio Video Interleave</i>) | Standar <i>container</i> untuk <i>file multimedia</i> dengan sistem operasi Windows |
| MP4 (dikenal sebagai .mp4) | Bentuk <i>container</i> standar untuk MPEG-4 |
| FLV (<i>Flash Video</i>) | Format yang digunakan untuk mengirimkan video MPEG melewati <i>Flash Player</i> |
| MOV | Format <i>container</i> untuk program Quicktime milik Apple |
| OGG, OGM dan OGV | <i>Container</i> dengan standar <i>opensource</i> |
| MKV (<i>Mastroska</i>) | <i>Container</i> lain dengan standar <i>opensource</i> yang biasa digunakan untuk anime |
| VOB (<i>DVD Video Object</i>) | <i>Container</i> standar untuk DVD |
| ASF | Format dari Microsoft untuk WMV dan WMA dengan akhiran .wmv atau .asf |

Sumber: shallowsky.com/linux/videoformats.html, 2014

Codec (singkatan dari “*coder/decoder*”) adalah perangkat untuk mengkodekan suara atau video menjadi aliran bit-bit data (*bytes*).

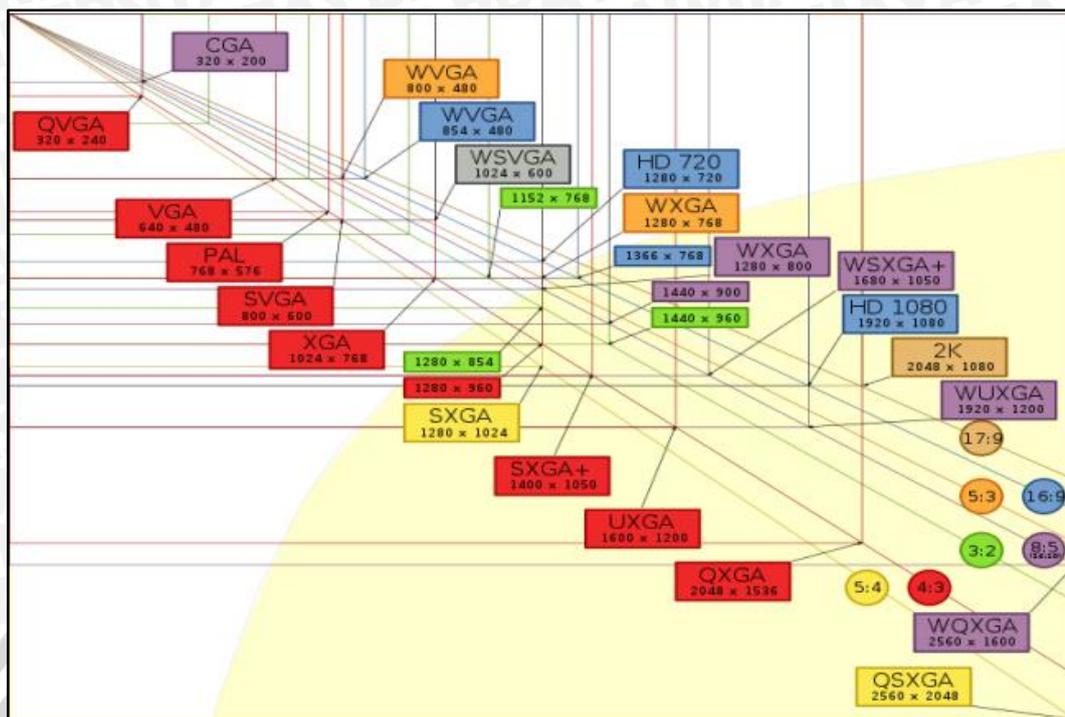
Tabel 2.2 Daftar Jenis *Codec Video*

| Format | ISO/IEC number Issue date | Target bandwidth (bit/s) | Resolusi | Aplikasi |
|---------------------------------|------------------------------|-----------------------------|---|--|
| H.261 | 1988-1990 | 384 k-2 M | 176 x 144 or 352 x 288 | <i>Video Conference, low delay</i> |
| H.263 | 1992 | 28,8 k-768 k | 128 x 96 to 720 x 480 | <i>Video Conference</i> |
| MPEG-1 | 11172 1993 | 400 k-1.5 M | 352 x 288 | CD-ROM |
| MPEG-2 | 13818 1994 | 1.5 M-15 M | 720 x 480 | Siaran Televisi, DVD |
| MPEG-4 | 14496 1998 | 28.8 K -500 K | 176 x 144 or 352 x 288 | Fixed dan <i>mobile web</i> |
| AVC, H.264/ MPEG-4 part 2 | 14496-10 2002 | 100 K – 500 K | 144p, 240p, 360p, 480p, 360p, 720p, 1080p | Keperluan umum, <i>Video On Demand</i> |

Sumber: David Austerberry, 2005

2.2.2 Parameter Video

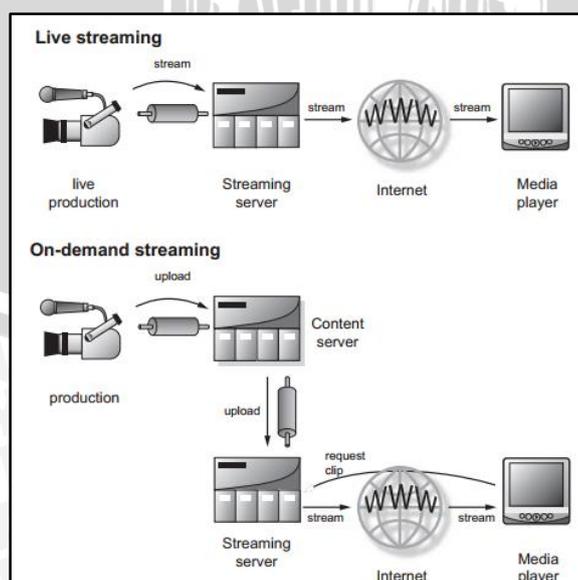
- 1) *Frames per Second (FPS)* – standar untuk FPS adalah 20 dan 30 *frame per second*. Semakin besar nilai FPS maka semakin banyak gambar per detik yang digunakan pada video maka pergerakan video lebih halus. Semakin kecil nilai FPS membuat video dengan pergerakan yang kaku.
- 2) *Video Bitrate – Bitrate* adalah nilai pengukuran dari bit yang dikirimkan per waktu tertentu. Nilai *bitrate* akhir/total adalah kombinasi dari *bitrate stream* video, *audio stream*, dengan nilai *bitrate stream* video yang mendominasi. Untuk 2 video dengan durasi yang sama, Semakin besar nilai *bitrate* suatu video maka semakin banyak informasi yang terkandung pada video tersebut dan semakin besar ukuran *file* tersebut.
- 3) *Resolution* – Resolusi menunjukkan ukuran gambar yang ditampilkan pada layar. Semakin tinggi resolusi semakin jernih gambar dan semakin besar *file* tersebut (shallowsky.com/linux, 2014).



Gambar 2.1 Resolusi Video
 Sumber: shallowsky.com/linux, 2014

2.2.3 Video Streaming

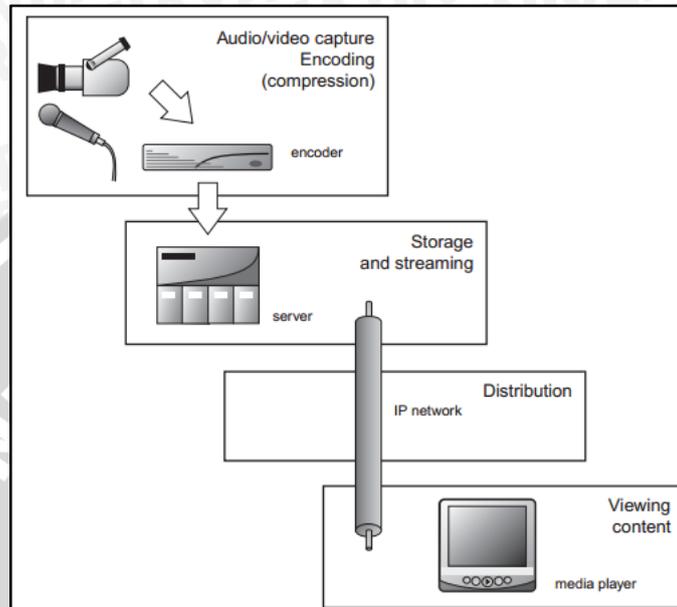
Video streaming adalah teknis mengirimkan gambar yang bergerak dan komponennya (suara) secara serial, data *streaming* dikompresi menjadi ukuran yang lebih kecil dari aslinya kemudian dikirimkan melalui jaringan internet dan ditampilkan oleh *media player* ketika *video* tersebut telah diterima oleh pengguna. Pengguna membutuhkan *media player*, yaitu aplikasi yang melakukan proses dekompresi, dan penampilan video ke layar monitor dan suara ke bagian *speaker*.



Gambar 2.2 Live dan On-Demand
 Sumber: David Austerberry, 2005

Video streaming terdiri dari dua jenis, yaitu:

1. *Live Streaming*, tayangan yang ditampilkan secara langsung (siaran langsung).
2. *VOD (Video On Demand)*, video yang ditampilkan sudah terlebih dahulu direkam (*pre-encoded*) atau disimpan dalam *server* seperti yang ditunjukkan pada Gambar 2.2 (David Austerberry, 2005).



Gambar 2.3 Arsitektur *Streaming*

Sumber: David Austerberry, 2005

Pada Gambar 2.3 ditunjukkan bahwa dalam arsitektur *streaming* terdapat empat buah komponen. Komponen-komponen tersebut ditunjukkan Tabel 2.3.

Tabel 2.3 Komponen *Streaming*

| No. | Komponen | Keterangan |
|-----|----------------------------------|--|
| 1 | <i>Capture dan Encoding</i> | Pada bagan ini terjadi proses <i>capture audio</i> dan video, kompresi media menjadi format tertentu |
| 2 | <i>Serving</i> | Bagian yang menunjukkan proses penyimpanan <i>file streaming</i> dan penyiaran pada pengguna |
| 3 | <i>Distribution dan Delivery</i> | Proses jalannya <i>file streaming</i> melalui jalur data yang telah terjalin koneksinya antara server dan pengguna |
| 4 | <i>Media Player</i> | Mendekodekan <i>file streaming</i> agar bisa dinikmati oleh pengguna sebagai <i>file audio</i> dan video |

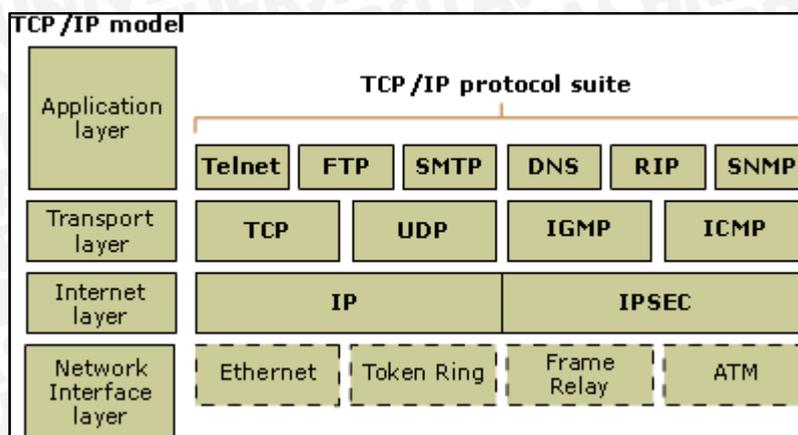
Sumber: David Austerberry, 2005

Faktor-faktor yang berpengaruh dalam distribusi *video streaming* melalui jaringan antara lain besar *delay* (waktu tunda), *lost packet*, dan juga teknik mendistribusikan *video* tersebut ke beberapa tujuan secara merata dan efisien (Apostolopoulos, 2001).

2.2.4 Protokol *Video Streaming*

Protokol adalah aturan yang mengatur komunikasi data antara 2 perangkat. Tanpa protokol, 2 perangkat yang tersambung tidak dapat berkomunikasi karena tidak ada aturan

komunikasi yang sama, sama halnya seperti seseorang yang berbahasa Prancis tidak dapat dipahami oleh orang yang berbahasa Jepang. (Behrouz Forouzan, 2007).



Gambar 2.4 Model TCP/IP
Sumber: technet.microsoft.com, 2005

Berdasarkan model jaringan TCP/IP pada Gambar 2.4, TCP/IP terdiri dari empat *layer*. Tipe-tipe layanan yang disediakan dan protokol-protokol yang digunakan pada masing-masing *layer* pada model TCP/IP dideskripsikan lebih detail pada Tabel 2.4.

Tabel 2.4 Layer-Layer TCP/IP

| Layer | Deskripsi | Protokol |
|-------------------|---|--|
| Application | Mendefinisikan protokol aplikasi-aplikasi dan bagaimana mengendalikan <i>interface</i> dari program dengan <i>transport layer</i> untuk menggunakan jaringan. | HTTP, Telnet, FTP, TFTP, dan lain-lain. |
| Transport | Menyediakan manajemen sesi komunikasi antara komputer-komputer. Mendefinisikan level layanan dan status koneksi jaringan saat dipakai. | TCP, UDP, RTP |
| Internet | Memaketkan data menjadi datagram-datagram IP, yang mengandung sumber dan alamat tujuan untuk digunakan proses <i>routing</i> . | IP, ICMP, ARP, RARP |
| Network Interface | Mengatur bagaimana data dikirimkan menuju jaringan secara fisik, termasuk bagaimana bit-bit diubah menjadi sinyal listrik oleh perangkat keras kemudian dikirimkan lewat media kabel <i>coaxial</i> , FO atau kabel tembaga <i>twisted-pair</i> . | Ethernet, <i>Token Ring</i> , FDDI, RS-232, X.25, <i>Frame Relay</i> |

Sumber: technet.microsoft.com, 2005

Streaming membutuhkan protokol transmisi yang mengabaikan data eror dan teknis retransmisi data (*unreliable*). Jika *streaming* melakukan teknis retransmisi data dan tidak mengabaikan data eror maka proses *streaming* akan memiliki *delay* yang relatif besar dan menyebabkan siaran menjadi berulang-ulang menuju *frame* yang dikirim ulang. Oleh karena itu pada *streaming* digunakan protokol UDP. Sifat UDP ini ditunjukkan oleh Tabel 2.5.

Tabel 2.5 TCP *versus* UDP

| TCP | UDP |
|----------------------------|------------------------|
| <i>Connection Oriented</i> | <i>Connectionless</i> |
| <i>Reliable</i> | <i>Unreliable</i> |
| <i>Controls Data Flow</i> | <i>No flow control</i> |

Sumber: David Austerberry, 2005

Selain menggunakan UDP, *streaming* juga menggunakan protokol RTP pada *layer transport*. RTP atau *Real Time Protocol* adalah protokol *transport* yang dikembangkan untuk kegiatan *streaming* data. RTP menyediakan *timestamp*, nomor urut untuk waktu kedatangan data. *Media player* pada pengguna akan menggunakan RTP sebagai pengatur paket yang diterima untuk diurutkan sesuai dengan langkah main. RTP selalu digunakan bersamaan dengan UDP. (David Austerberry, 2005).

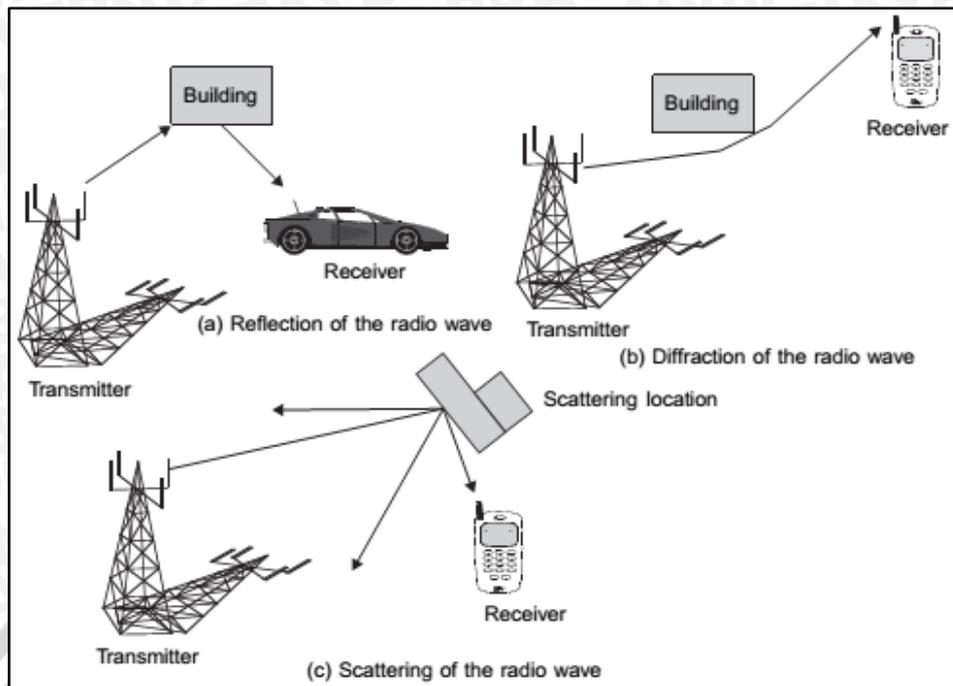
2.3 Propagasi Gelombang Radio

Gelombang radio merambat di udara sebagai gelombang elektromagnetik. Energi sinyal terbentuk dari medan listrik dan magnetik.

2.3.1 Mekanisme Dasar Propagasi Gelombang Radio

Reflection, *diffraction* dan *scattering* adalah 3 mekanisme dasar propagasi gelombang radio dan ditunjukkan pada Gambar 2.5.

- *Reflection*, atau refleksi, terjadi saat gelombang elektromagnetik yang merambat lalu menumbuk suatu obyek yang memiliki dimensi sangat luas dibandingkan panjang gelombang yang dirambatkan. Refleksi berasal dari permukaan bumi, bangunan dan dinding.
- *Diffraction*, atau difraksi, terjadi saat terdapat permukaan yang memiliki tepi (*edges*) tajam berada di antara lintasan radio pengirim dan penerima. Gelombang elektromagnetik yang dibiaskan dari permukaan ini mengalami pembengkokan.
- *Scattering*, atau hamburan, terjadi saat gelombang menabrak obyek-obyek yang berukuran kecil dibandingkan panjang gelombang. Obyek yang ditabrak berjumlah banyak. (Vijay K. Garg, 2007)



Gambar 2.5 Mekanisme Perambatan Gelombang
Sumber: Vijay K. Garg, 2007

2.3.2 Model Propagasi *Outdoor*

Model propagasi *outdoor* merupakan model propagasi luar ruangan. Sistem seluler pada dasarnya melingkupi daerah yang sangat luas dan terdapat beberapa penghalang yang menyebabkan redaman yang dapat mempengaruhi karakteristik propagasi *radio*. Profil lingkungan seluler yang diamati pada penelitian adalah wilayah urban. Daerah *urban* memiliki ciri-ciri antara lain:

- 1) Daerah pusat kota baik metropolis maupun kota menengah dengan gedung-gedung yang rapat dan tinggi.
- 2) Daerah yang terdapat perumahan, perkantoran dengan gedung-gedung yang terdiri dari lima sampai sepuluh lantai, hotel, dan rumah sakit
- 3) Gedung-gedung yang terdapat di daerah tersebut berkerangka logam dan beton tebal, sehingga membatasi propagasi *radio* melalui gedung.
- 4) Redaman oleh pepohonan diabaikan, karena pepohonan sangat jarang. (Ali Akhmadi, 2006)

2.3.2.1 Model Propagasi COST231

Model propagasi COST231-Walfisch Ikegami adalah model propagasi *outdoor* yang digunakan untuk memperkirakan nilai *pathloss* pada daerah dengan profil lingkungan seluler urban dan memiliki *range* frekuensi 800-2000 Mhz. Model propagasi ini menyempurnakan perhitungan *pathloss* Hata karena lebih banyak memasukkan besarnya nilai-nilai parameter yang menjadi karakteristik dari daerah urban, yang meliputi ketinggian gedung, lebar jalan,

dan jarak antar gedung. Perhitungan nilai *pathloss* dirumuskan dalam persamaan (2.1) (Vijay K. Garg, 2007).

$$L_{CWI} = L_{building} = L_{fs} + L_{rts} + L_{ms} \quad (2.1)$$

dengan:

$L_{CWI} = L_{building} = pathloss$ (dB)

L_{fs} = *free space loss* (dB)

L_{rts} = *rooftop to street diffraction* dan *scatter loss* (dB)

L_{ms} = *multiscreen diffraction loss* (dB)

Besarnya nilai L_{fs} dihitung dengan menggunakan persamaan (2.2) (Vijay K. Garg, 2007).

$$L_{fs} = 32,4 + 20 \log d + 20 \log f_c \quad (2.2)$$

dengan:

L_{fs} = *free space loss* (dB)

d = radius sel (km)

f_c = frekuensi *carrier* (MHz)

Besarnya nilai L_{rts} dihitung dengan menggunakan persamaan (2.3) (Vijay K. Garg, 2007).

$$L_{rts} = -16,9 - 10 \log w + 10 \log f_c + 20 \log \Delta h_m + L_o \quad (2.3)$$

dengan:

L_{rts} = *rooftop to street diffraction* dan *scatter loss* (dB)

w = lebar jalan (m)

f_c = frekuensi *carrier* (MHz)

Δh_m = $h_r - h_m$ (m)

h_r = tinggi gedung (m)

h_m = tinggi antena MS (m)

L_o = *street orientation function* (dB)

$$= -10 + 0,354\phi \quad \text{untuk } 0^\circ \leq \phi < 35^\circ$$

$$= 2,5 + 0,075 (\phi - 35^\circ) \quad \text{untuk } 35^\circ \leq \phi < 55^\circ$$

$$= 4,0 - 0,114 (\phi - 55^\circ) \quad \text{untuk } 55^\circ \leq \phi < 90^\circ$$

Besarnya nilai L_{ms} dihitung dengan menggunakan persamaan (2.4) (Vijay K. Garg, 2007).

$$L_{ms} = L_{bsh} + k_a + k_d \log d + k_f \log f_c - 9 \log b \quad (2.4)$$

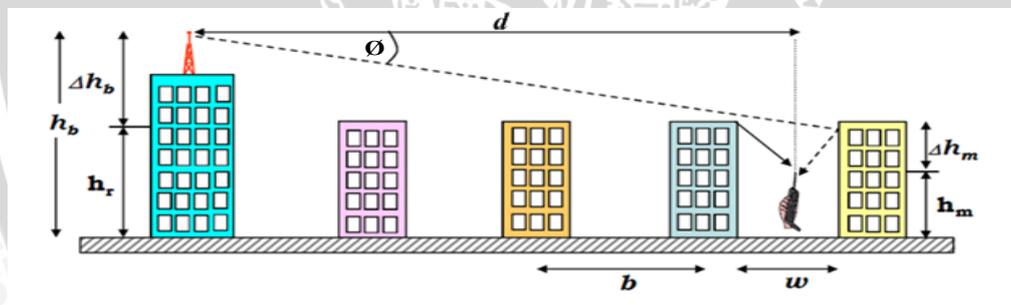
dengan:

L_{ms} = *multiscreen diffraction loss* (dB)

L_{bsh} = *shadowing gain (negative loss)* (dB)

$$= -18 \log (1 + \Delta h_b) \quad \text{untuk } h_b > h_r$$

- $= 0$ untuk $h_b \leq h_r$
 k_a = konstanta koreksi untuk tinggi antena
 $= 54$ untuk $h_b > h_r$
 $= 54 - 0,8 \Delta h_b$ untuk $d \geq 0,5$ km dan $h_b \leq h_r$
 $= 54 - 1,6 \Delta h_b \cdot d$ untuk $d < 0,5$ km dan $h_b \leq h_r$
 k_d = untuk mengendalikan ketergantungan L_{ms} terhadap jarak
 $= 18$ untuk $h_b > h_r$
 $= 18 - 15 \frac{\Delta h_b}{h_r}$ untuk $h_b \leq h_r$
 Δh_b = $h_b - h_r$ (m)
 h_b = tinggi antena *Base Station*
 d = jarak Tx - Rx (km)
 f_c = frekuensi *carrier* (MHz)
 k_f = untuk mengendalikan ketergantungan L_{ms} terhadap frekuensi
 $= 4 + 0,7 \left(\frac{f}{925} - 1 \right)$ untuk daerah urban dan suburban
 $= 4 + 1,5 \left(\frac{f}{925} - 1 \right)$ untuk daerah urban padat
 b = jarak antar gedung (m)



Gambar 2.6 Propagasi Gelombang Radio
 Sumber: Theodore S. Rappaport, 2002

2.4 High Speed Downlink Packet Access (HSDPA)

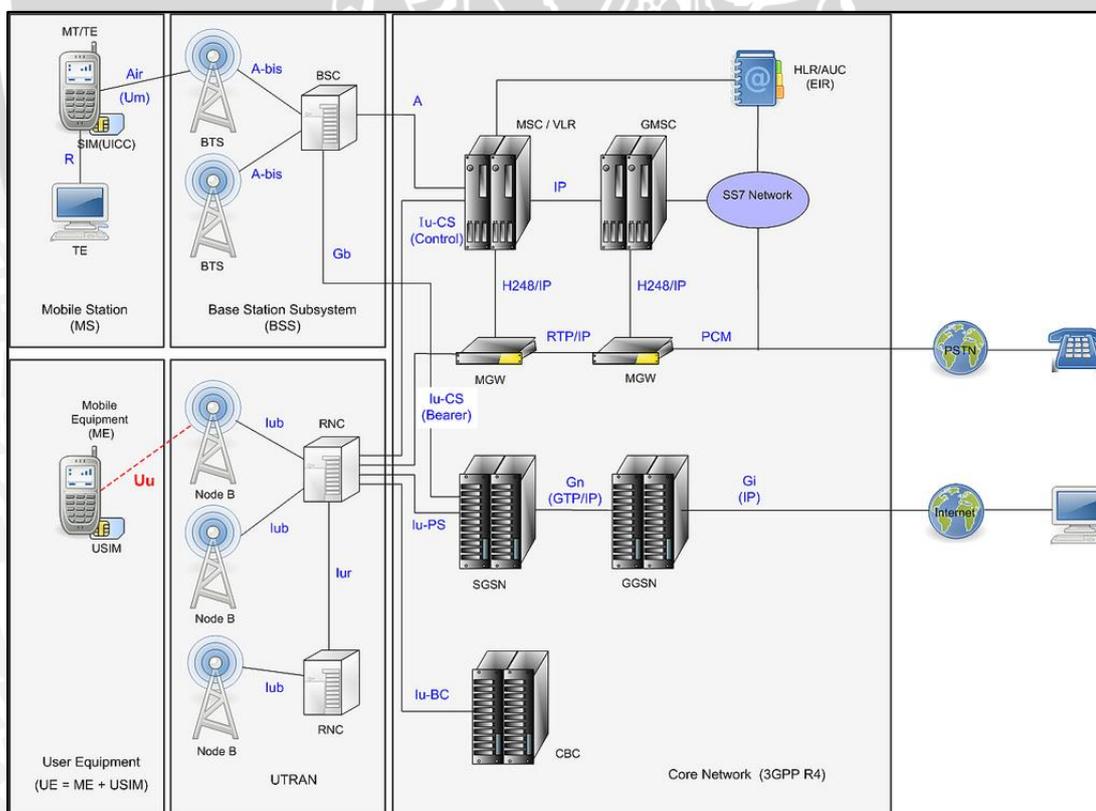
High Speed Downlink Packet Access atau disingkat HSDPA, adalah perkembangan dari jaringan radio WCDMA. HSDPA memiliki peningkatan performansi kanal *downlink* paket data WCDMA dan kemampuan untuk meninggikan data *rate* puncak, pengurangan *delay* dan peningkatan kapasitas. Hal ini dapat terjadi karena peningkatan teknik, yakni modulasi dengan orde tinggi, *rate control*, *channel dependent scheduling*, dan *hybrid ARQ with soft combining*.

Beberapa fitur yang digunakan HSDPA sehingga memiliki kelebihan untuk meningkatkan *throughput* dan mengurangi *delay end-to-end*, antara lain:

- 1) *High Speed - Downlink Shared Channel (HS-DSCH)*, kanal dengan kecepatan tinggi (hingga 13,98 Mbps) dan dapat digunakan bersamaan dengan pengguna lain.
- 2) *Transmission Time Interval (TTI)* yang pendek (2 ms), sehingga kecepatan transmisi pada *layer* fisik menjadi cepat.
- 3) Menggunakan teknik *Fast Scheduling*.
- 4) Menggunakan teknik *Adaptive Modulation and Coding (AMC)* memaksimalkan *data rate* dari *user equipment (UE)* sesuai kondisi kanal.
- 5) Menggunakan *fast Hybrid Automatic Repeat and Request (HARQ)*, meningkatkan performansi dan ketahanan terhadap *error* pada jalur (*link*) (Harri Holma & Antti Toskala, 2007).

2.4.1 Arsitektur HSDPA

HSDPA merupakan pengembangan dari jaringan UMTS, sehingga arsitektur jaringan HSDPA tetap menggunakan arsitektur jaringan UMTS. HSDPA adalah jaringan yang bersifat *packet switched network*. Secara umum arsitektur UMTS terdiri dari tiga bagian yaitu *user equipment (UE = ME + USIM)*, *access network (UTRAN)*, dan *core network*. Arsitektur UMTS secara sederhana ditunjukkan Gambar 2.7.



Gambar 2.7 Arsitektur Jaringan 3G/UMTS/WCDMA
Sumber: Clint Smith, 2006

1) Core Network (CN)

CN bertanggung jawab mengoneksikan UMTS dengan jaringan luar, menyediakan fungsi *switching* maupun *routing* panggilan untuk komunikasi suara (*circuit switching*), dan layanan *packet switched* untuk koneksi data. CN terdiri dari 2 bagian, yaitu:

- *Serving GPRS Support Node* (SGSN)

SGSN berfungsi sama halnya seperti *Mobile Switching Center/Visitor Location Register* (MSC/VLR), yaitu *database* data-data pelanggan yang tetap, antara lain: layanan utama pelanggan, layanan tambahan, serta mengenai lokasi pelanggan yang tetap. Secara khusus SGSN digunakan untuk layanan *Packet Switched* (PS).

- *Gateway GPRS Support Node* (GGSN)

GGSN berfungsi sebagai *interface* antara jaringan *backbone* PS dengan jaringan eksternal *Packet Data Network* (PDN), seperti jaringan internet.

2) Access Network

Access network bertanggung jawab dalam manajemen sumber *radio*. UMTS *Terrestrial Radio Access Network* (UTRAN) merupakan bagian dari *access network*. UTRAN merupakan bagian dari jaringan UMTS yang terdiri dari satu atau lebih *Radio Network Controller* (RNC) dan *Node B*. Semua yang terkait dengan fungsi *radio* dikontrol di dalam UTRAN. Sebuah UTRAN terkoneksi dengan jaringan eksternal ataupun UTRAN lain melalui CN. UTRAN terdiri dari dua bagian, yaitu:

- *Node B*

Node B merupakan unit fisik untuk mengirim atau menerima frekuensi pada sel. Tugas dari *Node B* adalah mengkonversi data antara *interface* Iub dan Uu.

- *Radio Network Controller* (RNC)

RNC dibagi menjadi dua, *Controlling RNC* dan *Serving RNC*. *Controlling RNC* bertanggung jawab dalam manajemen sumber *radio* yang tersedia pada *Node B*. Selama UE beroperasi, *Serving RNC* mengontrol sumber *radio* yang digunakan oleh UE dan mengakhiri *interface* Iu ke dan dari CN untuk layanan yang digunakan oleh UE.

3) *User Equipment (UE)*

UE merupakan terminal dari sisi pelanggan yang berhubungan dengan *radio interface* dari UTRAN dan aplikasi *user* serta dilengkapi dengan UMTS *Subscriber Identity Module (USIM)*. (Lingga Wardhana, 2011)

2.4.2 Kanal-Kanal pada HSDPA

Node B memperkirakan kualitas kanal dari setiap *user* yang sedang aktif pada jaringan HSDPA berdasarkan: jarak, *power control*, ACK/NACK *ratio*, QoS, dan CQI. *Node B* juga melakukan penjadwalan dan *link adaptation*, melakukan koneksi tergantung dari algoritma penjadwalan yang digunakan dan skema modulasi pada UE. Untuk dapat mengimplementasikan HSDPA, tiga kanal baru ditambahkan pada *platform WCDMA*. Operasi kanal HSDPA ditunjukkan Gambar 2.8. Beberapa kanal yang ditambahkan, antara lain:

1) *High Speed – Downlink Shared Channel (HS-DSCH)*

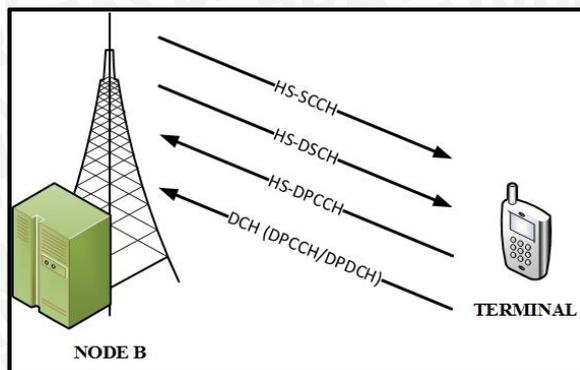
HS-DSCH sebagai kanal sharing untuk membawa beberapa *Dedicated Transport Channel (DCH)* dalam satu frekuensi. Kanal HS-DSCH mampu membawa data hingga mencapai *data rate* yang tinggi (13,98 Mbps).

2) *High Speed – Shared Control Channel (HS-SCCH)*

HS-SCCH digunakan untuk menandai jenis informasi sebelum penjadwalan TTI seperti *channelization code set*, skema modulasi, ukuran *transport block*, dan informasi protokol HARQ.

3) *High Speed - Uplink Dedicated Physical Control Channel (HS-DPCCH)*

HS-DPCCH bertanggung jawab dalam proses *uplink* yaitu pengiriman ACK dan *negative acknowledgement (NACK)* untuk memberikan status suatu paket data yang dikirim. Nilai bit digunakan untuk memilih skema modulasi dan *coding* yang sesuai untuk pengiriman selanjutnya, dari QPSK dengan turbo R=1/4 hingga 16-QAM dengan *turbo code* R=3/4, termasuk memilih untuk tidak melakukan pengiriman jika kondisi kanal buruk (Harri Holma & Antti Toskala, 2006).



Gambar 2.8 Kanal pada HSDPA
 Sumber: Holma dan Antti Toskala, 2006

2.4.3 Parameter Kinerja Jaringan

Quality of Service (QoS) adalah ukuran kualitas layanan yang digunakan untuk menyatakan kinerja jaringan berdasarkan parameter. Sedangkan menurut ITU-T E.800, QoS adalah performansi yang menentukan derajat kepuasan pengguna terhadap *service* yang diberikan oleh jaringan berdasarkan parameter-parameter. Pada jaringan HSDPA dan penelitian, QoS ditentukan dari parameter *delay end-to-end*, *throughput* dan *packet loss* yang dilihat dari sisi pengguna.

2.4.3.1 Delay Codec Video On Demand

Aplikasi *video on demand* pada penelitian menggunakan *codec* audio dan video pada *file video on demand* yang ditransmisikan. Spesifikasi *codec* audio dan video yang digunakan pada penelitian ditunjukkan Tabel 2.6.

Tabel 2.6 Spesifikasi Audio dan Video Codec

| Video CODEC | Bit Rate (kbps) | Maximum Payload (byte) | Delay CODEC (ms) |
|-----------------|-----------------|------------------------|------------------|
| AVC/H.264 MPEG4 | 96-384 | 254 | 10-50 |
| Audio CODEC | Bit Rate (kbps) | Maximum Payload (byte) | Delay CODEC (ms) |
| AAC | 128 | 63 | 3-60 |

Sumber: RFC 3640, 2003, dan RFC 3984, 2005

Besarnya nilai *delay codec* pada aplikasi *video on demand* penelitian ditunjukkan persamaan 2.5.

$$t_{codec} = t_{audio} + t_{video} \tag{2.5}$$

Keterangan:

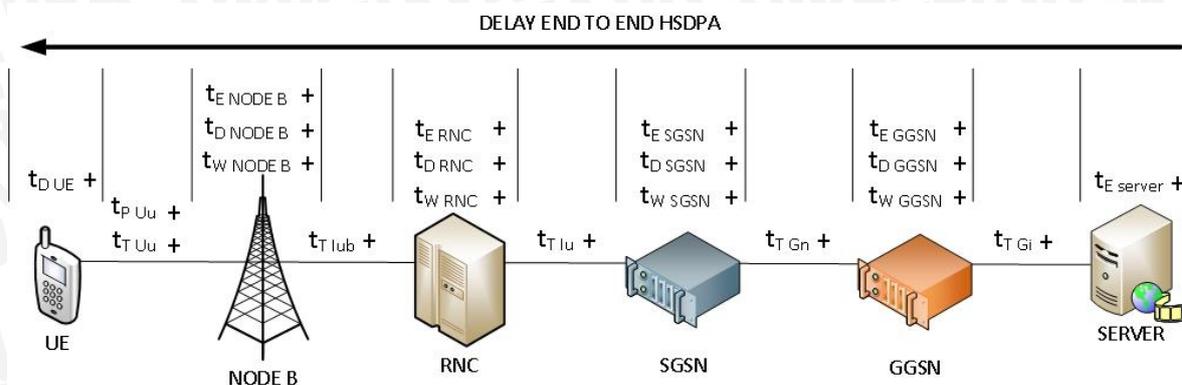
t_{codec} = Delay codec keseluruhan (s)

t_{audio} = Delay codec audio (s)

t_{video} = Delay codec video (s)

2.4.3.2 Delay End-to-End pada HSDPA

Delay pada jaringan HSDPA merupakan penjumlahan *delay-delay* dari ujung hingga ujung (*server hingga user equipment*). Dengan mengasumsikan paket data berbasis *World Wide Web* yang di-download dari *server* maka besarnya *delay end-to-end* diilustrasikan dalam Gambar 2.9.



Gambar 2.9 Delay End-to-End pada HSDPA
Sumber: Harri Holma & Antti Toskala, 2006

Penelitian membatasi perhitungan pada kanal *downlink*, yakni dari server menuju UE, karena aplikasi *video on demand* bersifat *broadcasting* artinya paket data video disiarkan server menuju UE tanpa UE mengirim siaran balik menuju server. Sehingga, *Delay End-to-End* berdasarkan Gambar 2.9 dihitung dengan menggunakan persamaan 2.6. (Harri Holma & Antti Toskala, 2006)

$$t_{\text{end to end}} = t_{\text{proses}} + t_t + t_p + t_w \quad (2.6)$$

Keterangan:

- $t_{\text{end to end}}$ = Delay end-to-end pada jaringan HSDPA (s)
- t_{proses} = Delay proses, yakni delay enkapsulasi dan dekapsulasi (s)
- t_t = Delay transmisi (s)
- t_p = Delay propagasi (s)
- t_w = Delay antrian (s)

a) Delay Proses

Delay proses adalah *delay* yang terjadi saat paket dienkapsulasi dan didekapsulasi pada *node*. Enkapsulasi adalah proses pemaketan data dengan menambahkan *header* pada paket data tersebut, sehingga paket data dapat dikirimkan tepat ke tujuan. Delay enkapsulasi adalah rentang waktu yang diperlukan suatu *node* untuk melakukan proses enkapsulasi. Sedangkan *delay* dekapsulasi adalah rentang waktu yang diperlukan suatu *node* untuk proses dekapsulasi atau membaca *header* dari paket yang diterima..

Besarnya *delay* dekapulasi dan *delay* enkapsulasi diperoleh menggunakan persamaan 2.7 dan 2.8 (Onno W. Purbo, 1998):

$$t_{\text{enc}} = \frac{W_{\text{frame}} - L}{c} \times 8 \quad (2.7)$$

$$t_{\text{dec}} = \frac{W_{\text{frame}} - L}{c} \times 8 \quad (2.8)$$

Keterangan:

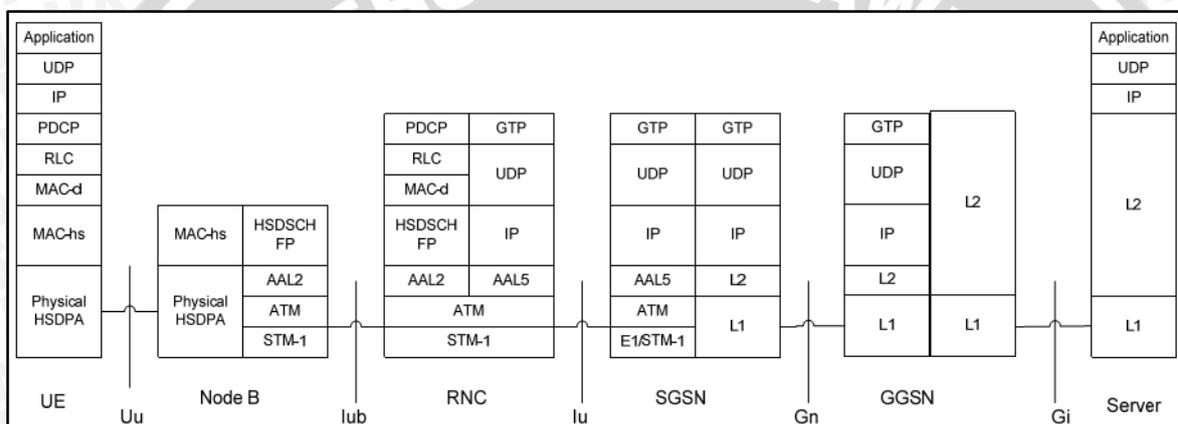
t_{enc} = *Delay* enkapsulasi (s)

t_{dec} = *Delay* dekapulasi (s)

W_{frame} = Panjang *frame* (byte)

L = Panjang paket data di *node* (byte)

C = Kecepatan transmisi data pada *node* (bps)



Gambar 2.10 Analisis *Delay* Enkapsulasi dan Dekapsulasi
Sumber: Harri Holma & Antti Toskala, 2006

Delay enkapsulasi jaringan HSDPA terjadi pada titik server, GGSN, SGSN, RNC, dan *Node B* sesuai Gambar 2.10. Sedangkan *delay* dekapulasi jaringan HSDPA terjadi pada titik GGSN, SGSN, RNC, *Node B* dan UE. Besarnya *delay* enkapsulasi pada HSDPA ditunjukkan persamaan 2.9 (Harri Holma & Antti Toskala, 2006).

$$t_{\text{enc}} = t_{\text{e-server}} + t_{\text{e-GGSN}} + t_{\text{e-SGSN}} + t_{\text{e-RNC}} + t_{\text{e-NodeB}} \quad (2.9)$$

Keterangan:

t_{enc} = *Delay* enkapsulasi total (s)

$t_{\text{e-server}}$ = *Delay* enkapsulasi server (IP network) (s)

$t_{\text{e-GGSN}}$ = *Delay* enkapsulasi GGSN (s)

$t_{\text{e-SGSN}}$ = *Delay* enkapsulasi SGSN (s)

$t_{\text{e-RNC}}$ = *Delay* enkapsulasi RNC (s)

$t_{\text{e-NodeB}}$ = *Delay* enkapsulasi *Node-B* (s)

Besarnya *delay* dekapulasi pada HSDPA ditunjukkan persamaan 2.10 (Harri Holma & Antti Toskala, 2006).

$$t_{dec} = t_{d-GGSN} + t_{d-SGSN} + t_{d-RNC} + t_{d-NodeB} + t_{d-UE} \quad (2.10)$$

Keterangan:

| | |
|---------------|--------------------------------|
| t_{dec} | = Delay dekapsulasi total (s) |
| t_{d-GGSN} | = Delay dekapsulasi GGSN (s) |
| t_{d-SGSN} | = Delay dekapsulasi SGSN (s) |
| t_{d-RNC} | = Delay dekapsulasi RNC (s) |
| $t_{d-NodeB}$ | = Delay dekapsulasi Node-B (s) |
| t_{d-UE} | = Delay dekapsulasi UE (s) |

Sehingga besarnya nilai *delay* proses pada jaringan HSDPA ditunjukkan persamaan 2.11 (Harri Holma & Antti Toskala, 2006).

$$t_{proses} = t_{enc} + t_{dec} \quad (2.11)$$

- **Server**

Server berfungsi sebagai sumber siaran VOD, paket data video disiarkan dari server menuju UE. Dalam perhitungan saat data mencapai *transport layer*, data dibentuk menjadi beberapa segmen data. Pada penelitian ini protokol yang digunakan pada *Network Layer* adalah UDP. Paket data *video on demand* ditambahkan *header* RTP, UDP, NALU dan IP pada layer *Transport*.

Untuk menghitung besar *payload* video dan *payload* audio pada *video streaming* (*video on demand*) ditentukan dengan persamaan 2.12 dan 2.13. (Wes Simpson, 2008)

$$P_v = B_{CODEC-V} \times \text{frame size video} \quad (2.12)$$

$$P_a = B_{CODEC-A} \times \text{frame size audio} \quad (2.13)$$

Keterangan:

| | |
|-------------------------|--|
| P_v | = Besarnya paket data video pada <i>file</i> (bit) |
| P_a | = Besarnya paket data audio pada <i>file</i> (bit) |
| $B_{CODEC-V}$ | = Besar kecepatan <i>codec</i> video (bps) |
| $B_{CODEC-A}$ | = Besar kecepatan <i>codec</i> audio (bps) |
| <i>Frame Size</i> video | = Lama waktu 1 <i>frame</i> video (s) |
| <i>Frame Size</i> audio | = Lama waktu audio untuk 1 <i>frame</i> video (s) |

Jumlah *payload* yang disegmentasi berdasarkan *payload* maksimum dari ketentuan *codec*. Sehingga, jumlah paket data video dan audio sesuai dengan persamaan 2.14 dan 2.15 (Wes Simpson, 2008).

$$N_v = \frac{P_v}{P_v \text{ maksimum} \times 8} \quad (2.14)$$

$$N_a = \frac{P_a}{P_a \text{ maksimum} \times 8} \quad (2.15)$$

Keterangan:

N_v = Jumlah paket video

N_a = Jumlah paket audio

P_v maksimum = *Payload* video maksimum berdasarkan *codec* (byte)

P_a maksimum = *Payload* audio maksimum berdasarkan *codec* (byte)

Payload video on demand (video dan audio) masing-masing ditambahkan *header* NALU, RTP, UDP dan IPv4. Besarnya paket video dan audio setelah dienkodakan dan ditambah *header* sesuai dengan persamaan 2.16 dan 2.17 (Wes Simpson, 2008).

$$P_v(\text{encoded}) = P_v + N_v \times (H_{\text{NALU}} + H_{\text{RTP}} + H_{\text{UDP}} + H_{\text{IPv4}}) \quad (2.16)$$

Dan untuk audio,

$$P_a(\text{encoded}) = P_a + N_a \times (H_{\text{NALU}} + H_{\text{RTP}} + H_{\text{UDP}} + H_{\text{IPv4}}) \quad (2.17)$$

Keterangan:

$P_v(\text{encoded})$ = *Payload* video setelah penambahan *header* NALU, RTP, UDP, IP (byte)

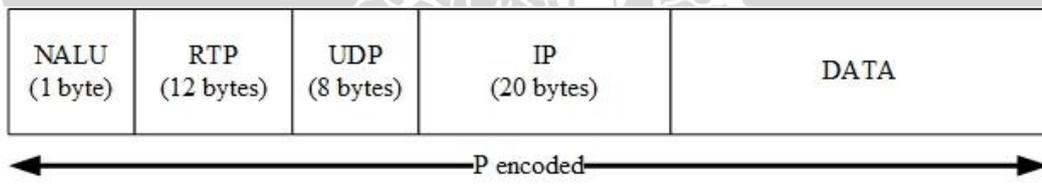
$P_a(\text{encoded})$ = *Payload* audio setelah penambahan *header* NALU, RTP, UDP, IP (byte)

H_{NALU} = *Header* NALU (byte)

H_{RTP} = *Header* RTP (byte)

H_{UDP} = *Header* UDP (byte)

H_{IPv4} = *Header* IPv4 (byte)



Gambar 2.11 Format Datagram UDP RTP *Payload*

Besarnya paket data aplikasi *video on demand* yang ditransmisikan pada sistem ditentukan dengan persamaan 2.18 (Wes Simpson, 2008).

$$P_{\text{vod size}} = P_v(\text{encoded}) + P_a(\text{encoded}) \quad (2.18)$$

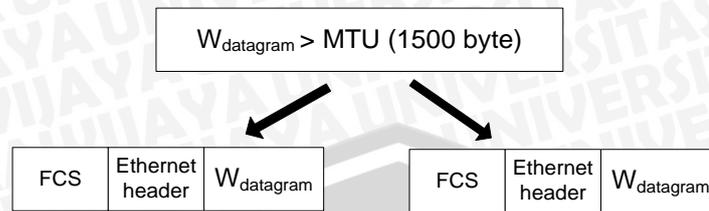
Keterangan:

$P_{\text{vod size}}$ = Panjang datagram VOD (byte)

$P_v(\text{encoded})$ = *Payload* video setelah penambahan *header* NALU, RTP, UDP, IP (byte)

$P_a(\text{encoded})$ = *Payload* audio setelah penambahan *header* NALU, RTP, UDP, IP (byte)

Dari *transport layer* data dibawa menuju ke *network layer*. IP diubah menjadi datagram. Jika datagram IP tidak melebihi MTU Ethernet, maka datagram IP langsung ditambahkan dengan *header* Ethernet dan FCS (Heywood, 1997).



Gambar 2.12 Format Segmentasi Datagram IP
Sumber: Taufan Riza, 2002

Jika datagram IP melebihi *Maximum Transmission Unit* (MTU) Ethernet sebesar 1500 byte, maka datagram IP tersebut akan disegmentasi terlebih dahulu baru ditambahkan dengan *header* Ethernet dan FCS sesuai dengan persamaan 2.19 (Heywood, 1997).

$$N_{\text{frame Ethernet}} = \frac{P_{\text{VOD-size}}}{\text{MTU}_{\text{Ethernet}}} \quad (2.19)$$

Keterangan:

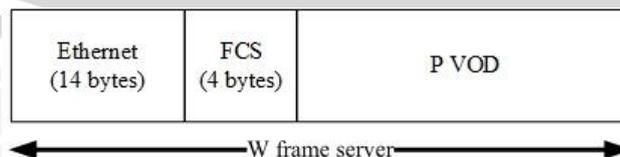
- $N_{\text{frame Ethernet}}$ = Jumlah *frame* Ethernet
- $P_{\text{VOD-size}}$ = Panjang datagram VOD (byte)
- $\text{MTU}_{\text{Ethernet}}$ = MTU Ethernet (1500 byte)

Maka, besar *frame* yang dikirimkan server menuju GGSN ditunjukkan persamaan 2.20 (Harri Holma & Antti Toskala, 2006).

$$W_{\text{frame server}} = P_{\text{VOD-size}} + [N_{\text{frame Ethernet}} \times (H_{\text{Ethernet}} + \text{FCS})] \quad (2.20)$$

Keterangan:

- $W_{\text{frame server}}$ = Panjang *frame server* (byte)
- $P_{\text{vod size}}$ = Panjang datagram VOD (byte)
- $N_{\text{frame Ethernet}}$ = Jumlah *frame* Ethernet
- H_{Ethernet} = *Header* Ethernet (byte)
- FCS = *Frame Check Sequence* (byte)



Gambar 2.13 *Frame Server*

Sehingga *delay* enkapsulasi pada server dapat ditentukan dengan persamaan 2.21.

$$t_{e\text{-server}} = \frac{W_{\text{frame server}} - P_{\text{VOD-size}}}{C_{\text{server}}} \times 8 \quad (2.21)$$

Keterangan:

- $t_{e-server}$ = Delay enkapsulasi server (s)
 $W_{frame server}$ = Panjang *frame server* (byte)
 $P_{vod size}$ = Panjang datagram VOD (byte)
 C_{server} = Kecepatan transmisi data pada server (bps)

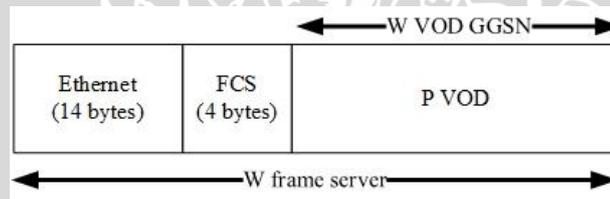
- **Gateway GPRS Support Node (GGSN)**

Pada GGSN, paket data yang diterima dari server mengalami proses dekapsulasi dan enkapsulasi untuk dikirim ke *node* selanjutnya. Paket data yang didekapsulasi pada GGSN ditunjukkan oleh persamaan 2.22 (Harri Holma & Antti Toskala, 2006).

$$W_{VOD\ GGSN} = W_{frame\ server} - H_{Ethernet} - FCS \quad (2.22)$$

Keterangan:

- $W_{VOD\ GGSN}$ = Panjang paket data *video on demand* di GGSN (byte)
 $W_{frame\ server}$ = Panjang *frame server* (byte)
 $H_{Ethernet}$ = Header Ethernet (byte)
 FCS = Frame Check Sequence (byte)



Gambar 2.14 Paket Data VOD GGSN

Besarnya *delay* dekapsulasi yang terjadi di titik GGSN ditunjukkan oleh persamaan 2.23.

$$t_{d-GGSN} = \frac{W_{frame\ sever} - W_{VOD\ GGSN}}{C_{GGSN}} \times 8 \quad (2.23)$$

Keterangan:

- t_{d-GGSN} = Delay dekapsulasi GGSN (s)
 $W_{frame\ server}$ = Besar *frame server* (byte)
 $W_{VOD\ GGSN}$ = Panjang paket data *video on demand* di GGSN (byte)
 C_{GGSN} = Kecepatan transmisi data pada GGSN (bps)

Paket data yang telah didekapsulasi dan diterima oleh GGSN, selanjutnya dienkapsulasi kembali untuk dikirim menuju *node* SGSN. Pada GGSN terdapat ukuran maksimum *frame*, yakni MSS. MSS adalah *Maximum Segment Size*. MSS adalah ukuran terbesar paket data dalam satuan byte yang memiliki nilai kurang dari MTU. Paket data pada jaringan melewati banyak *node* dan router, maka MSS membantu paket data agar

tidak *oversize*. Nilai MSS pada GGSN didapatkan dari persamaan 2.24 (Harri Holma & Antti Toskala, 2006).

$$MSS_{GGSN} = MTU_{Ethernet} - H_{GTP} - H_{UDP} - H_{IPv4} \quad (2.24)$$

Keterangan:

- MSS_{GGSN} = *Maximum Segment Size* (byte)
 $MTU_{Ethernet}$ = MTU Ethernet (1500 byte)
 H_{GTP} = *Header GPRS Tunneling Protocol* (GTP) (byte)
 H_{UDP} = *Header User Datagram Protocol* (UDP) (byte)
 H_{IPv4} = *Header Internet Protocol* versi 4 (IP) (byte)

Paket data *video on demand* pada GGSN disegmentasi dengan MSS, jika paket data tersebut melebihi ukuran MSS. Paket data yang disegmentasi oleh MSS ditentukan dengan persamaan 2.25 (Harri Holma & Antti Toskala, 2006).

$$N_{datagram} = \frac{W_{VOD\ GGSN}}{MSS_{GGSN}} \quad (2.25)$$

Keterangan:

- $N_{datagram}$ = Jumlah *frame* MSS pada GGSN
 $W_{VOD\ GGSN}$ = Panjang paket data *video on demand* di GGSN (byte)
 MSS_{GGSN} = *Maximum Segment Size* (byte)

Paket data *video on demand* pada GGSN dienkapsulasi kembali menjadi datagram dengan penambahan *header* GTP, UDP dan IP sesuai dengan persamaan 2.26 (Harri Holma & Antti Toskala, 2006).

$$W_{datagram\ GGSN} = W_{VOD\ GGSN} + N_{datagram} \times (H_{GTP} + H_{UDP} + H_{IPv4}) \quad (2.26)$$

Keterangan:

- $W_{datagram\ GGSN}$ = Panjang datagram GGSN (byte)
 $N_{datagram}$ = Jumlah *frame* MSS pada GGSN
 $W_{VOD\ GGSN}$ = Panjang paket data *video on demand* di GGSN (byte)
 H_{GTP} = *Header GPRS Tunneling Protocol* (GTP) (byte)
 H_{UDP} = *Header User Datagram Protocol* (UDP) (byte)
 H_{IPv4} = *Header Internet Protocol* versi 4 (IP) (byte)

Untuk dikirimkan menuju SGSN, maka datagram GGSN disegmentasi kembali jika melebihi MTU (1500 byte). Persamaan segmentasi datagram GGSN dengan MTU *Ethernet* ditunjukkan oleh persamaan 2.27 (Harri Holma & Antti Toskala, 2006).

$$N_{frame\ Ethernet\ GGSN} = \frac{W_{datagram\ GGSN}}{MTU_{Ethernet}} \quad (2.27)$$

Keterangan:

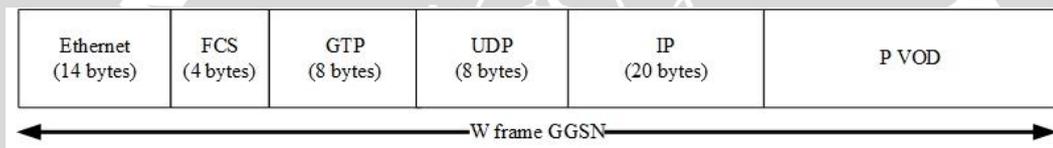
$N_{frame \text{ Ethernet GGSN}}$ = Jumlah *frame* Ethernet GGSN
 $W_{datagram \text{ GGSN}}$ = Panjang datagram GGSN (byte)
 MTU_{Ethernet} = MTU Ethernet (1500 byte)

Sehingga jumlah keseluruhan *frame* GGSN yang dikirim menuju SGSN dapat diketahui dengan persamaan 2.28 (Harri Holma & Antti Toskala, 2006).

$$W_{frame \text{ GGSN}} = W_{datagram \text{ GGSN}} + [N_{frame \text{ Ethernet GGSN}} \times (H_{\text{Ethernet}} + FCS)] \quad (2.28)$$

Keterangan:

$W_{frame \text{ GGSN}}$ = Panjang *frame* GGSN (byte)
 $W_{datagram \text{ GGSN}}$ = Panjang datagram GGSN (byte)
 $N_{frame \text{ Ethernet GGSN}}$ = Jumlah *frame* Ethernet GGSN
 H_{Ethernet} = *Header* Ethernet (byte)
 FCS = *Frame Check Sequence* (byte)



Gambar 2.15 *Frame Data* pada GGSN untuk dikirim menuju SGSN

Sehingga *delay* enkapsulasi yang terjadi pada GGSN ditunjukkan dengan persamaan 2.29.

$$t_{e\text{-GGSN}} = \frac{W_{frame \text{ GGSN}} - W_{VOD \text{ GGSN}}}{C_{\text{GGSN}}} \times 8 \quad (2.29)$$

Keterangan:

$t_{e\text{-GGSN}}$ = *Delay* enkapsulasi GGSN (s)
 $W_{frame \text{ GGSN}}$ = Panjang *frame* GGSN (byte)
 $W_{VOD \text{ GGSN}}$ = Panjang paket data *video on demand* di GGSN (byte)
 C_{GGSN} = Kecepatan transmisi data pada GGSN (bps)

- **Serving GPRS Support Node (SGSN)**

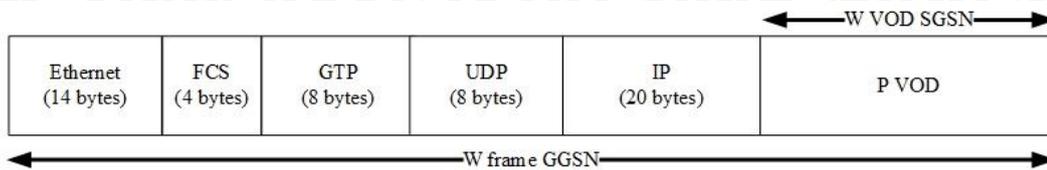
Paket data yang diterima dari GGSN didekapsulasi selanjutnya dienkapsulasi untuk dikirimkan menuju *node* berikutnya. Paket data yang diterima pada SGSN dari GGSN ditunjukkan dengan persamaan 2.30 (Harri Holma & Antti Toskala, 2006).

$$W_{VOD \text{ SGSN}} = W_{frame \text{ GGSN}} - H_{\text{GTP}} - H_{\text{UDP}} - H_{\text{IPv4}} - H_{\text{Ethernet}} - FCS \quad (2.30)$$

Keterangan:

$W_{VOD \text{ SGSN}}$ = Panjang paket data *video on demand* di SGSN (byte)
 $W_{frame \text{ GGSN}}$ = Panjang *frame* GGSN (byte)

| | |
|----------------|--|
| H_{GTP} | = Header GPRS Tunneling Protocol (GTP) (byte) |
| H_{UDP} | = Header User Datagram Protocol (UDP) (byte) |
| H_{IPv4} | = Header Internet Protocol versi 4 (IP) (byte) |
| $H_{Ethernet}$ | = Header Ethernet (byte) |
| FCS | = Frame Check Sequence (byte) |



Gambar 2.17 Paket Data VOD SGSN

Besarnya *delay* dekapsulasi pada SGSN didapatkan dengan persamaan 2.31.

$$t_{d-SGSN} = \frac{W_{frame\ GGSN} - W_{VOD\ SGSN}}{C_{SGSN}} \times 8 \quad (2.31)$$

Keterangan:

| | |
|-------------------|--|
| t_{d-SGSN} | = <i>Delay</i> dekapsulasi SGSN (s) |
| $W_{frame\ GGSN}$ | = Panjang <i>frame</i> GGSN (byte) |
| $W_{VOD\ SGSN}$ | = Panjang paket data <i>video on demand</i> di SGSN (byte) |
| C_{SGSN} | = Kecepatan transmisi data pada SGSN (bps) |

Paket data yang telah didekapsulasi dan diterima oleh SGSN, selanjutnya didekapsulasi kembali untuk dikirim menuju *node* RNC. Pada SGSN terdapat ukuran maksimum *frame*, yakni MSS. Nilai MSS didapatkan dari persamaan 2.32 (Harri Holma & Antti Toskala, 2006).

$$MSS_{SGSN} = MTU - H_{GTP} - H_{UDP} - H_{IPv4} \quad (2.32)$$

Keterangan:

| | |
|------------------|--|
| MSS_{SGSN} | = <i>Maximum Segment Size</i> pada SGSN (byte) |
| $MTU_{Ethernet}$ | = MTU Ethernet (1500 byte) |
| H_{GTP} | = Header GPRS Tunneling Protocol (GTP) (byte) |
| H_{UDP} | = Header User Datagram Protocol (UDP) (byte) |
| H_{IPv4} | = Header Internet Protocol versi 4 (IP) (byte) |

Datagram *video on demand* pada SGSN disegmentasi dengan MSS, jika datagram tersebut melebihi ukuran MSS. Paket data yang disegmentasi oleh MSS ditentukan dengan persamaan 2.33 (Harri Holma & Antti Toskala, 2006).

$$N_{datagram\ SGSN} = \frac{W_{VOD\ SGSN}}{MSS_{SGSN}} \quad (2.33)$$

Keterangan:

$$N_{\text{frame ATM}} = \frac{W_{\text{AAL5}}}{48 \text{ byte}} \quad (2.36)$$

Keterangan:

$N_{\text{frame ATM}}$ = Jumlah *frame* ATM

W_{AAL5} = Panjang *frame* AAL5 (byte)

Setiap *frame* ATM diberi *header* (H_{ATM}) sebesar 5 byte sehingga panjang *frame* ATM menjadi 53 byte. Panjang *frame* yang siap ditransmisikan menuju GGSN merupakan panjang *frame* ATM total sesuai dengan persamaan 2.37 (Harri Holma & Antti Toskala, 2006).

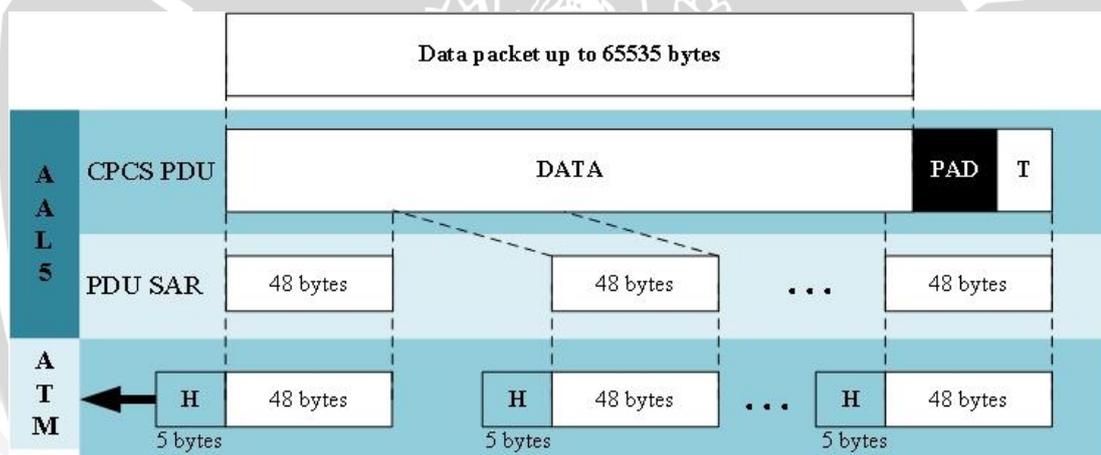
$$W_{\text{frame SGSN}} = N_{\text{frame ATM}} \times W_{\text{frame ATM}} \quad (2.37)$$

Keterangan:

$W_{\text{frame SGSN}}$ = Panjang *frame* SGSN (byte)

$N_{\text{frame ATM}}$ = Jumlah *frame* ATM

$W_{\text{frame ATM}}$ = Panjang *frame* ATM (byte)



Gambar 2.18 Format *Frame* AAL5

Sumber: Behrouz A. Forouzan, 2007

Sehingga *delay* enkapsulasi yang terjadi pada SGSN ditunjukkan oleh persamaan 2.38.

$$t_{e\text{-SGSN}} = \frac{W_{\text{frame SGSN}} - W_{\text{VOD SGSN}}}{C_{\text{SGSN}}} \times 8 \quad (2.38)$$

Keterangan:

$t_{e\text{-SGSN}}$ = *Delay* enkapsulasi SGSN (s)

$W_{\text{frame SGSN}}$ = Panjang *frame* SGSN (byte)

$W_{\text{VOD SGSN}}$ = Panjang paket data *video on demand* di SGSN (byte)

C_{SGSN} = Kecepatan transmisi data pada SGSN (bps)

- **Radio Network Controller (RNC)**

Paket data diterima oleh RNC dan diproses dekapsulasi dan enkapsulasi. Proses dekapsulasi dilakukan terlebih dahulu saat paket data dari SGSN diterima oleh RNC. Paket data yang diterima RNC dari SGSN ditunjukkan dengan persamaan 2.39 (Harri Holma & Antti Toskala, 2006).

$$W_{\text{VOD RNC}} = W_{\text{frame SGSN}} - H_{\text{GTP}} - H_{\text{UDP}} - H_{\text{IPv4}} - H_{\text{AAL5}} - H_{\text{ATM}} \quad (2.39)$$

Keterangan:

$W_{\text{VOD RNC}}$ = Panjang paket data *video on demand* di RNC (byte)

$W_{\text{frame SGSN}}$ = Panjang *frame* SGSN (byte)

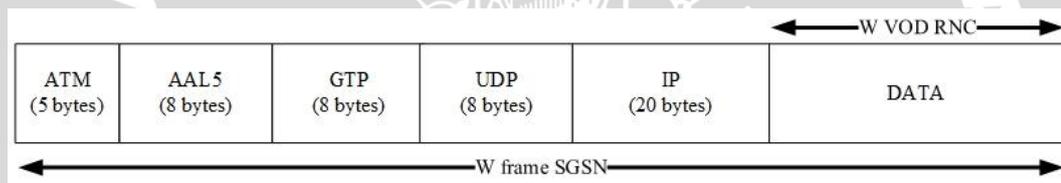
H_{GTP} = *Header GPRS Tunneling Protocol* (GTP) (byte)

H_{UDP} = *Header User Datagram Protocol* (UDP) (byte)

H_{IPv4} = *Header Internet Protocol* versi 4 (IP) (byte)

H_{AAL5} = *Header AAL5/CPCS-PDU* pada SGSN (byte)

H_{ATM} = *Header ATM* pada SGSN (byte)



Gambar 2.19 Paket Data VOD RNC

Sistem transmisi *Synchronous Transport Module -1* (STM-1) merupakan *interface* sistem transmisi yang digunakan pada RNC. STM-1 memiliki kecepatan data sebesar 155,52 Mbps. Sehingga nilai *delay* dekapsulasi pada RNC dapat ditentukan dengan persamaan 2.40.

$$t_{d\text{-RNC}} = \frac{W_{\text{frame SGSN}} - W_{\text{VOD RNC}}}{C_{\text{RNC}}} \times 8 \quad (2.40)$$

Keterangan:

$t_{d\text{-RNC}}$ = *Delay* dekapsulasi RNC (s)

$W_{\text{frame SGSN}}$ = Panjang *frame* SGSN (byte)

$W_{\text{VOD RNC}}$ = Panjang paket data *video on demand* di RNC (byte)

C_{RNC} = Kecepatan transmisi data RNC (bps)

Packet Data Convergence Protocol (PDCP) adalah *layer* RNC yang berfungsi mengenkapsulasi datagram IP *video on demand* pada RNC. Persamaan untuk enkapsulasi *video on demand* adalah persamaan 2.41 (Harri Holma & Antti Toskala, 2006).

$$N_{\text{datagram RNC}} = \frac{W_{\text{VOD RNC}}}{\text{MTU Ethernet}} \quad (2.41)$$

Keterangan:

$N_{\text{datagram RNC}}$ = Jumlah paket datagram RNC

$W_{\text{VOD RNC}}$ = Panjang paket data *video on demand* di RNC (byte)

$\text{MTU}_{\text{Ethernet}}$ = MTU Ethernet (1500 byte)

Frame PDCP yang terbentuk adalah *payload* data ditambah *header* PDCP. Besar *frame* total pada *layer* PDCP dapat ditentukan dengan persamaan 2.42 (Harri Holma & Antti Toskala, 2006).

$$W_{\text{frame PDCP}} = W_{\text{VOD RNC}} + (N_{\text{datagram RNC}} \times H_{\text{PDCP}}) \quad (2.42)$$

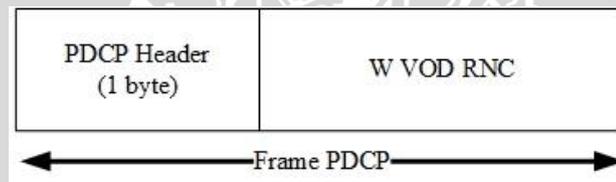
Keterangan:

$W_{\text{frame PDCP}}$ = Panjang *frame* PDCP (byte)

$W_{\text{VOD RNC}}$ = Panjang paket data *video on demand* di RNC (byte)

$N_{\text{datagram RNC}}$ = Jumlah paket datagram RNC

H_{PDCP} = *Header* PDCP (byte)



Gambar 2.20 *Frame* PDCP

Selanjutnya pada *layer Radio Link Control* (RLC), *frame* PDCP disegmentasi menjadi RLC PDU *fixed size* sebesar 40 byte (ETSI TS 125-322 v6.3.0). Sehingga persamaan untuk menentukan jumlah *frame* RLC ditunjukkan persamaan 2.43 (Harri Holma & Antti Toskala, 2006).

$$N_{\text{frame RLC}} = \frac{W_{\text{frame PDCP}}}{40 \text{ byte}} \quad (2.43)$$

Keterangan:

$N_{\text{frame RLC}}$ = Jumlah *frame* pada RLC

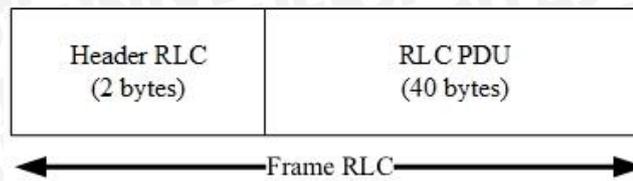
$W_{\text{frame PDCP}}$ = Panjang *frame* PDCP (byte)

Panjang setiap *frame* RLC adalah sebanyak 42 byte karena setiap *frame* selanjutnya diberi *header* sebanyak 2 byte (H_{RLC}). Panjang *frame* RLC total yang diteruskan menuju *layer Medium Access Control-dedicated* (MAC-d) ditentukan dengan persamaan 2.44 (Harri Holma & Antti Toskala, 2006).

$$W_{\text{frame RLC total}} = N_{\text{frame RLC}} \times W_{\text{frame RLC}} \quad (2.44)$$

Keterangan:

$W_{\text{frame RLC total}}$ = Panjang *frame* RLC total (byte)
 $N_{\text{frame RLC}}$ = Jumlah *frame* pada RLC
 $W_{\text{frame RLC}}$ = Panjang *frame* RLC (byte)

Gambar 2.21 *Frame* RLC

Saat di *layer* MAC-d, RLC PDU disegmentasi menjadi MAC-d *Service Data Unit* (MAC-d SDU) sebesar 42 byte. Jumlah *frame* MAC-d SDU ditentukan dengan persamaan 2.45 (Harri Holma & Antti Toskala, 2006).

$$N_{\text{frame MAC-d}} = \frac{W_{\text{frame RLC total}}}{42 \text{ byte}} \quad (2.45)$$

Keterangan:

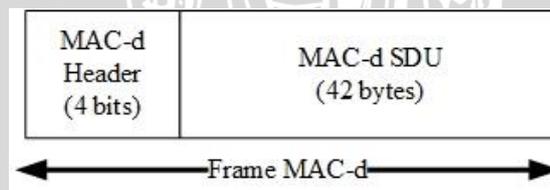
$N_{\text{frame MAC-d}}$ = Jumlah *frame* MAC-d
 $W_{\text{frame RLC total}}$ = Panjang *frame* RLC total (byte)

Sedangkan panjang *frame* MAC-d ditentukan dengan persamaan 2.46 (Erik Dahlman, 2008).

$$W_{\text{frame MAC-d}} = H_{\text{MAC-d}} + \text{MAC-d SDU} \quad (2.46)$$

Keterangan:

$W_{\text{frame MAC-d}}$ = Panjang *frame* MAC-d (byte)
 $H_{\text{MAC-d}}$ = *Header* MAC-d (byte)
 MAC-d SDU = *Payload* MAC-d (byte)

Gambar 2.22 *Frame* MAC-d

Frame MAC-d ditambahkan *header Frame Protocol* (FP) dan *Cyclic Redundancy Check* (CRC) untuk dilakukan enkapsulasi pada kanal *High Speed Downlink Shared Channel-Frame Protocol* (HS-DSCH FP). Persamaannya ditunjukkan pada persamaan 2.47 (Harri Holma & Antti Toskala, 2006).

$$W_{\text{frame FP}} = W_{\text{frame MAC-d}} + H_{\text{FP}} + \text{CRC} \quad (2.47)$$

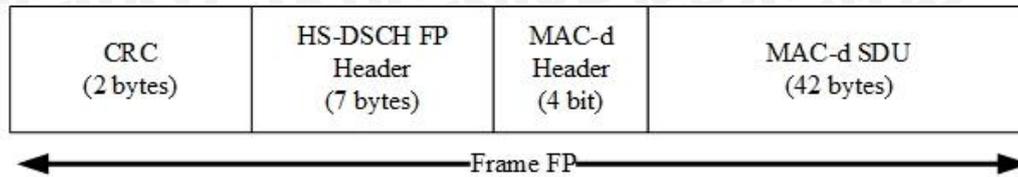
Keterangan:

$W_{\text{frame FP}}$ = Panjang *frame* HSDSCH-FP (byte)

$W_{\text{frame MAC-d}}$ = Panjang *frame* MAC-d (byte)

H_{FP} = *Header* HSDSCH-FP (byte)

CRC = *Cyclic Redundancy Check* (byte)



Gambar 2.23 *Frame FP*

Panjang *frame* HS-DSCH FP total diperoleh melalui persamaan 2.48 (Harri Holma & Antti Toskala, 2006).

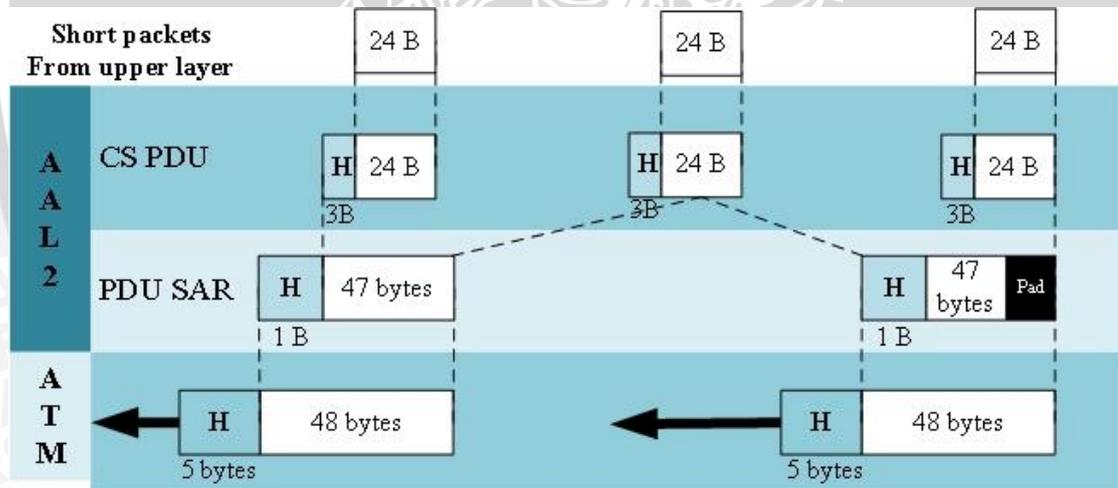
$$W_{\text{frame FP total}} = N_{\text{frame MAC-d}} \times W_{\text{frame FP}} \quad (2.48)$$

Keterangan:

$W_{\text{frame FP total}}$ = Panjang *frame* HSDSCH-FP total (byte)

$N_{\text{frame MAC-d}}$ = Jumlah *frame* MAC-d

$W_{\text{frame FP}}$ = Panjang *frame* HSDSCH-FP (byte)



Gambar 2.24 *Frame AAL2*

Sumber: Behrouz A. Forouzan, 2007

Saat paket data masuk menuju lapisan AAL2 seperti yang ditunjukkan Gambar 2.22, paket data dibentuk menjadi *Common Part Sublayer* (CPS) dan blok-blok PDU SAR. Persamaan untuk menentukan CS PDU ditunjukkan persamaan 2.49 (Behrouz A. Forouzan, 2007).

$$N_{\text{CS-PDU}} = \frac{W_{\text{frame FP total}}}{24 \text{ byte}} \quad (2.49)$$

Keterangan:

$N_{\text{CS-PDU}}$ = Jumlah paket data CS-PDU

$W_{frame\ FP\ total}$ = Panjang *frame* HSDSCH-FP total (byte)

Untuk menentukan besar paket data total pada CS-PDU dapat diperoleh dengan menggunakan persamaan 2.50 (Behrouz A. Forouzan, 2007).

$$W_{frame\ CS-PDU\ total} = W_{frame\ FP\ total} + (N_{CS-PDU} \times H_{CS-PDU}) \quad (2.50)$$

Keterangan:

$W_{frame\ CS-PDU\ total}$ = Panjang paket data CS-PDU total (byte)

$W_{frame\ FP\ total}$ = Panjang *frame* HSDSCH-FP total (byte)

N_{CS-PDU} = Jumlah paket data CS-PDU

H_{CS-PDU} = *Header* CS-PDU (byte)

Setelah menentukan nilai besar paket data total, *frame* CS-PDU dibagi menjadi blok-blok *payload* PDU (*Segmentation and Reassembly Sublayer*) atau PDU SAR sebesar 47 byte dan ditunjukkan oleh persamaan 2.51 (Behrouz A. Forouzan, 2007).

$$N_{PDU-SAR} = \frac{W_{frame\ CS-PDU\ total}}{47\ byte} \quad (2.51)$$

Keterangan:

$N_{PDU-SAR}$ = Jumlah *frame* PDU-SAR

$W_{frame\ CS-PDU\ total}$ = Panjang paket data CS-PDU total (byte)

Blok PDU SAR ditambahkan *header* ($H_{PDU-SAR}$) sebesar 1 byte sehingga besar tiap blok PDU SAR menjadi sebesar 48 byte. Persamaan untuk menentukan panjang paket data PDU SAR diperoleh dengan menggunakan persamaan 2.52 (Behrouz A. Forouzan, 2007).

$$W_{frame\ PDU\ SAR\ total} = N_{PDU-SAR} \times 48\ byte \quad (2.52)$$

Keterangan:

$W_{frame\ PDU-SAR\ total}$ = Panjang *frame* PDU-SAR total (byte)

$N_{PDU-SAR}$ = Jumlah *frame* PDU-SAR

Frame AAL2 dipecah menjadi *fixed cell* ATM dengan besar 48 byte pada *layer* ATM. Sehingga jumlah *frame* ATM yang terbentuk dapat ditentukan dengan persamaan 2.53 (Behrouz A. Forouzan, 2007).

$$N_{frame\ ATM} = \frac{W_{frame\ PDU-SAR\ total}}{48\ byte} \quad (2.53)$$

Keterangan:

$N_{frame\ ATM}$ = Jumlah *frame* ATM pada RNC

$W_{frame\ PDU-SAR\ total}$ = Panjang *frame* PDU-SAR total (byte)

Setiap *frame* ATM diberi *header* (H_{ATM}) sebesar 5 byte. Panjang *frame* yang ditransmisikan menuju *Node-B* merupakan panjang *frame* ATM total dan ditunjukkan dengan persamaan 2.54 (Behrouz A. Forouzan, 2007).

$$W_{frame\ RNC} = N_{frame\ ATM} \times W_{frame\ ATM} \quad (2.54)$$

Keterangan:

| | |
|--------------------------|--|
| $W_{frame\ RNC}$ | = Panjang <i>frame</i> RNC (byte) |
| $N_{frame\ ATM}$ | = Jumlah <i>frame</i> ATM pada RNC |
| $W_{frame\ ATM\ single}$ | = Panjang <i>frame</i> ATM per paket (53 byte) |

Sehingga besar *delay* dekapulasi yang terjadi pada RNC dapat ditentukan dengan persamaan 2.55.

$$t_{e-RNC} = \frac{W_{frame\ RNC} - W_{VOD\ RNC}}{C_{RNC}} \times 8 \quad (2.55)$$

Keterangan:

| | |
|------------------|---|
| t_{e-RNC} | = <i>Delay</i> enkapsulasi RNC (s) |
| $W_{frame\ RNC}$ | = Panjang <i>frame</i> RNC (byte) |
| $W_{VOD\ RNC}$ | = Panjang paket data <i>video on demand</i> di RNC (byte) |
| C_{RNC} | = Kecepatan transmisi data pada RNC (bps) |

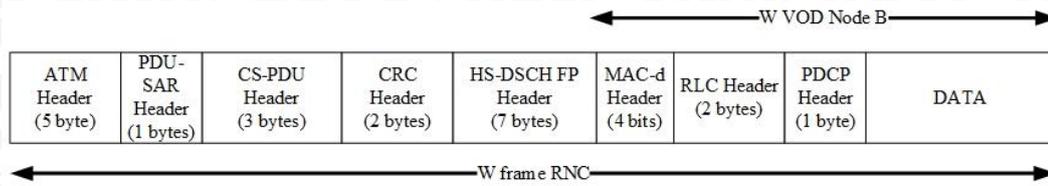
- **Node B**

Paket data yang diterima oleh *Node B* dari RNC mengalami proses dekapulasi dan selanjutnya mengalami proses enkapsulasi untuk dikirim menuju UE. Paket data yang diterima *Node B* mengalami proses dekapulasi dengan persamaan 2.56 (Harri Holma & Antti Toskala, 2006).

$$W_{VOD\ Node-B} = W_{frame\ RNC} - H_{ATM} - H_{CS-PDU} - H_{PDU-SAR} - H_{FP} - CRC \quad (2.56)$$

Keterangan:

| | |
|-------------------|--|
| $W_{VOD\ Node-B}$ | = Panjang paket data <i>video on demand</i> <i>Node-B</i> (byte) |
| $W_{frame\ RNC}$ | = Panjang <i>frame</i> RNC (byte) |
| H_{ATM} | = <i>Header</i> ATM pada RNC (byte) |
| H_{CS-PDU} | = <i>Header</i> CS-PDU pada RNC (byte) |
| $H_{PDU-SAR}$ | = <i>Header</i> PDU-SAR pada RNC (byte) |
| H_{FP} | = Panjang <i>header</i> HSDSCH-FP (byte) |
| CRC | = <i>Cyclic Redundancy Check</i> (byte) |

Gambar 2.25 Format Paket Data pada *Node B*

Node B menggunakan *interface* STM-1 dengan kecepatan pemrosesan 155,52 Mbps. Sehingga *delay* dekapsulasi pada *Node B* dapat ditentukan dengan persamaan 2.57.

$$t_{d-Node\ B} = \frac{W_{frame\ RNC} - W_{VOD\ Node-B}}{C_{Node\ B}} \times 8 \quad (2.57)$$

Keterangan:

- $t_{d-Node\ B}$ = *Delay* dekapsulasi *Node B* (s)
 $W_{frame\ RNC}$ = Panjang *frame RNC* (byte)
 $W_{VOD\ Node-B}$ = Panjang paket data *video on demand* pada *Node-B* (byte)
 $C_{node\ B}$ = Kecepatan transmisi data di *Node B* (bps)

Pada *layer* MAC-hs, *frame* MAC PDU dan *header* MAC-hs (H_{MAC-hs}) dibentuk menjadi satu blok, yakni *Transport Block Size* (TBS) dengan besar 7298 bit (HSDPA by Siemens, 2004). Jumlah blok TBS yang dikirimkan menuju UE ditunjukkan dengan persamaan 2.58 (Harri Holma & Antti Toskala, 2006).

$$N_{TBS} = \frac{W_{VOD\ Node-B}}{7298\ bit} \times 8 \quad (2.58)$$

Keterangan:

- N_{TBS} = Jumlah blok TBS
 $W_{VOD\ Node-B}$ = Panjang paket data *video on demand* pada *Node-B* (byte)

Setiap blok TBS ditambahkan *header* CRC sebesar 2 byte sebelum dikirimkan menuju UE. Persamaannya ditunjukkan oleh persamaan 2.59 (Harri Holma & Antti Toskala, 2006).

$$W_{frame\ Node-B} = W_{VOD\ Node\ B} + N_{TBS} \times CRC \quad (2.59)$$

Keterangan:

- $W_{frame\ Node-B}$ = Panjang *frame Node B* atau MAC-hs (byte)
 $W_{VOD\ Node-B}$ = Panjang paket data *video on demand* pada *Node-B* (byte)
 N_{TBS} = Jumlah blok TBS
 CRC = *Cyclic Redundancy Check* (2 byte)

Delay enkapsulasi pada *Node B* dapat ditentukan dengan persamaan 2.60.

$$t_{e-Node\ B} = \frac{W_{frame\ Node-B} - W_{VOD\ Node-B}}{C_{Node-B}} \times 8 \quad (2.60)$$

Keterangan:

- $t_{e\text{-Node B}}$ = Delay enkapsulasi Node B (s)
 $W_{frame\ Node\text{-B}}$ = Panjang frame Node B atau MAC-hs (byte)
 $W_{VOD\ Node\text{-B}}$ = Panjang paket data *video on demand* pada Node-B (byte)
 $C_{node\ B}$ = Kecepatan transmisi data Node B (bps)

- **User Equipment (UE)**

Pada *user equipment* terjadi proses dekapsulasi. Proses dekapsulasi memproses paket data dari Node B dan dimodelkan dengan persamaan 2.61 (Harri Holma & Antti Toskala, 2006).

$$W_{VOD\ UE} = W_{frame\ Node\text{-B}} - H_{RTP} - H_{UDP} - H_{IPv4} - H_{PDCP} - H_{RLC} - H_{MAC\text{-d}} - H_{MAC\text{-hs}} - CRC \quad (2.61)$$

Keterangan:

- $W_{VOD\ UE}$ = Panjang paket data *video on demand* pada UE (byte)
 $W_{frame\ Node\text{-B}}$ = Panjang frame Node B atau MAC-hs (byte)
 H_{RTP} = Header RTP (byte)
 H_{UDP} = Header UDP (byte)
 H_{IPv4} = Header IPv4 (byte)
 H_{PDCP} = Header PDCP (byte)
 H_{RLC} = Header RLC (byte)
 $H_{MAC\text{-d}}$ = Header MAC-d (byte)
 $H_{MAC\text{-hs}}$ = Header MAC-hs (byte)
 CRC = Cyclic Redundancy Check (2 byte)

Maka, besarnya *delay* dekapsulasi pada UE dapat ditentukan dengan persamaan 2.62.

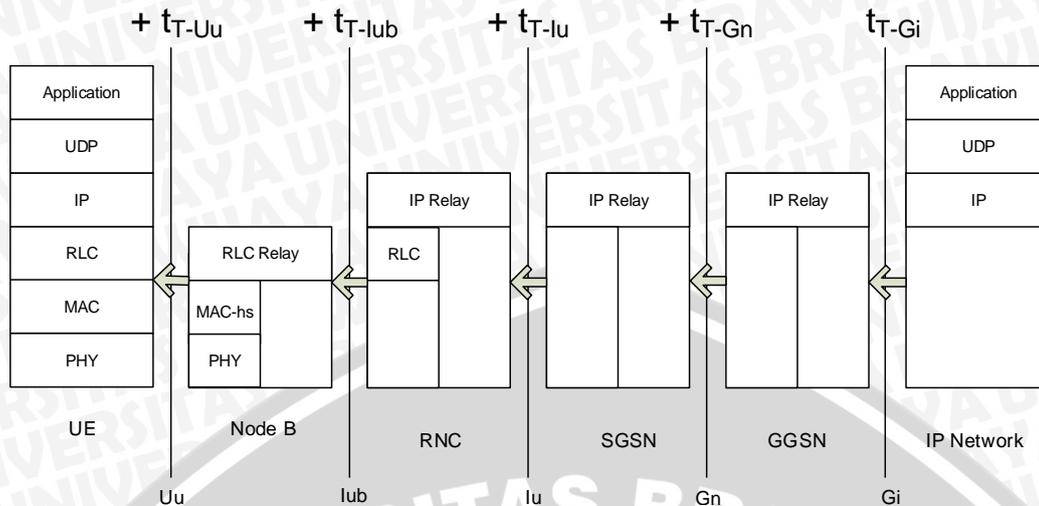
$$t_{d\text{-UE}} = \frac{W_{frame\ Node\text{-B}} - W_{VOD\ UE}}{C_{UE}} \times 8 \quad (2.62)$$

Keterangan:

- $t_{d\text{-Node B}}$ = Delay dekapsulasi UE (s)
 $W_{VOD\ UE}$ = Panjang paket data *video on demand* pada UE (byte)
 $W_{frame\ Node\text{-B}}$ = Panjang frame Node B atau MAC-hs (byte)
 C_{UE} = Kecepatan transmisi data UE (bps)

b) Delay Transmisi

Delay transmisi adalah waktu yang diperlukan paket data untuk melalui media transmisi dari satu *node* ke *node* yang lain. Media transmisi bisa melalui serat optik, jaringan lokal tembaga maupun *air interface*.



Gambar 2.26 Interface antar Node HSDPA
Sumber: Harri Holma & Antti Toskala, 2006

Berdasarkan Mischa Swartz, *delay* transmisi ditentukan dengan persamaan 2.63 (Mischa Schwartz, 1987).

$$t_T = \frac{W \times 8}{C} \quad (2.63)$$

Keterangan:

- t_T = *Delay* transmisi (s)
 W = Panjang *frame* pada suatu *node* (byte)
 C = Kecepatan transmisi (bps)

Sehingga *delay* transmisi total pada HSDPA sesuai Gambar 2.27, yakni meliputi *interface* Gi, Gn, lu, lub, Uu dapat ditentukan dengan persamaan 2.64

$$t_{T \text{ total}} = t_{T \text{ Gi}} + t_{T \text{ Gn}} + t_{T \text{ lu}} + t_{T \text{ lub}} + t_{T \text{ Uu}} \quad (2.64)$$

Keterangan:

- $t_{T \text{ total}}$ = *Delay* transmisi total (s)
 $t_{T \text{ Gi}}$ = *Delay* transmisi antara Server-GGSN(s)
 $t_{T \text{ Gn}}$ = *Delay* transmisi antara GGSN-SGSN (s)
 $t_{T \text{ lu}}$ = *Delay* transmisi antara SGSN-RNC(s)
 $t_{T \text{ lub}}$ = *Delay* transmisi antara RNC-Node B(s)
 $t_{T \text{ Uu}}$ = *Delay* transmisi antara Node B-UE(s)

c) *Delay Propagasi*

Menurut Forouzan dalam bukunya *Data Communications and Networking* tahun 2000, *delay* propagasi adalah waktu yang dibutuhkan oleh paket data dalam bentuk gelombang elektromagnetik untuk merambat dengan media udara dari *Node B* menuju UE. Persamaannya ditentukan oleh persamaan 2.65 (Behrouz A. Forouzan, 2000).

$$t_p = \frac{N_{\text{frame}} \times R}{c} \times 8 \quad (2.65)$$

Keterangan:

t_p = Delay propagasi (s)

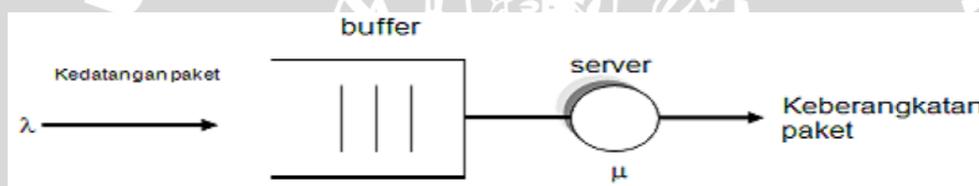
N_{frame} = Jumlah *frame* yang ditransmisikan

R = Jarak atau radius sel (m)

c = Cepat rambat gelombang elektromagnetik (3×10^8 m/s)

d) Delay Antrian

Delay antrian adalah rentang waktu paket data berada dalam antrian untuk ditransmisikan. Model antrian yang digunakan adalah model *First In First Out* (FIFO), yakni data pertama yang masuk akan diproses terlebih dahulu dan akan dikeluarkan terlebih dahulu. *Delay* antrian dihitung dengan menggunakan model antrian M/M/1. M pertama menunjukkan kedatangan *Poisson*, M kedua berarti distribusi waktu pelayanan eksponensial, dan 1 menunjukkan bahwa jumlah *node* yang melayani adalah tunggal. Disiplin antrian ditunjukkan pada Gambar 2.27.



Gambar 2.27 M/M/1

Sumber: Mischa Schwartz, 1987

Delay antrian pada jaringan HSDPA terjadi pada titik GGSN, SGSN, *Node B* dan RNC. (Sirtufillaila, 2010). Besarnya *delay* antrian tersebut ditentukan oleh persamaan 2.66 hingga 2.70. (Mischa Schwartz, 1987)

$$t_w = t_{\text{queue}} + t_{\text{serv}} \quad (2.66)$$

sedangkan,

$$t_{\text{queue}} = \frac{\lambda}{\mu(\mu - \lambda)} \quad (2.67)$$

dan,

$$t_{\text{serv}} = \frac{1}{\mu} \quad (2.68)$$

dan,

$$\mu = \frac{c}{L} \quad (2.69)$$

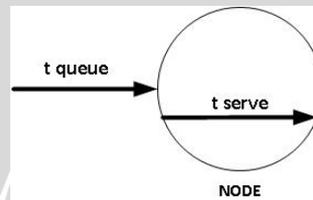
dan,

$$\lambda = \rho \cdot \mu \quad (2.70)$$

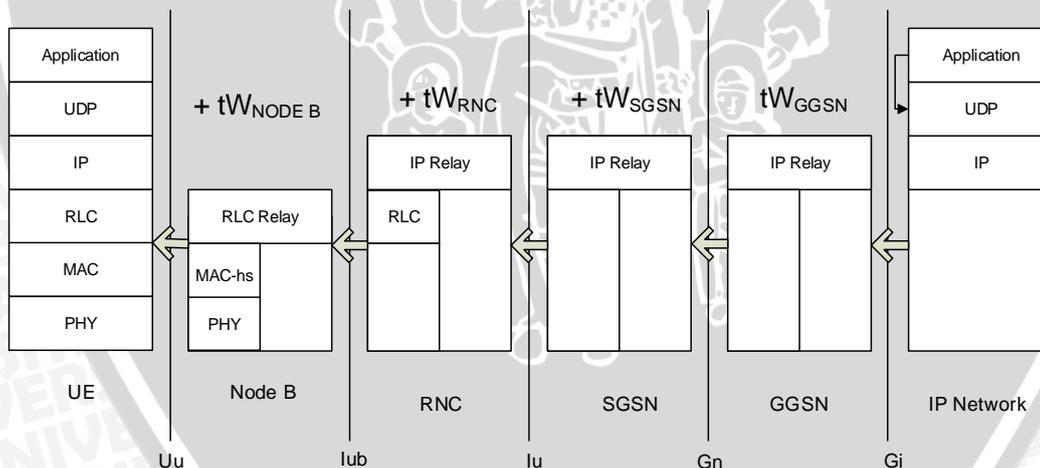
Keterangan:

- t_w = Delay antrian (s)
 t_{queue} = Waktu tunggu paket pada *node* (s)
 t_{serv} = Waktu rata-rata pelayanan *node* (s)
 λ = Kecepatan kedatangan paket pada *node* (bps)
 μ = Kecepatan pelayanan *node* (bps)
 C = Kecepatan transmisi (bps)
 L = Panjang paket data di *node* (bit)
 ρ = Faktor utilitas ($0 < \rho < 1$)

Faktor utilitas bernilai lebih besar dari 0 dan lebih kecil dari 1. Hal ini dikarenakan jika $\rho > 1$ berarti rata-rata kedatangan melampaui rata-rata pelayanan, sehingga panjang antrian bertambah tanpa batas.



Gambar 2.28 Waktu Tunggu Pelayanan (t_{serve}) dan Tunggu (t_{queue}) Paket



Gambar 2.29 Delay Antrian HSDPA

Sumber: Harri Holma & Antti Toskala, 2006

Sehingga *delay* antrian total pada HSDPA dapat ditentukan dengan persamaan 2.71.

$$t_w \text{ total} = t_w \text{ GGSN} + t_w \text{ SGSN} + t_w \text{ RNC} + t_w \text{ Node B} \quad (2.71)$$

Keterangan:

- $t_w \text{ total}$ = Delay antrian total (s)
 $t_w \text{ GGSN}$ = Delay antrian pada GGSN (s)
 $t_w \text{ SGSN}$ = Delay antrian pada SGSN (s)

$t_{w \text{ RNC}}$ = Delay antrian pada RNC (s)

$t_{w \text{ Node B}}$ = Delay antrian pada Node B (s)

e) Delay Total Sistem VOD Penelitian

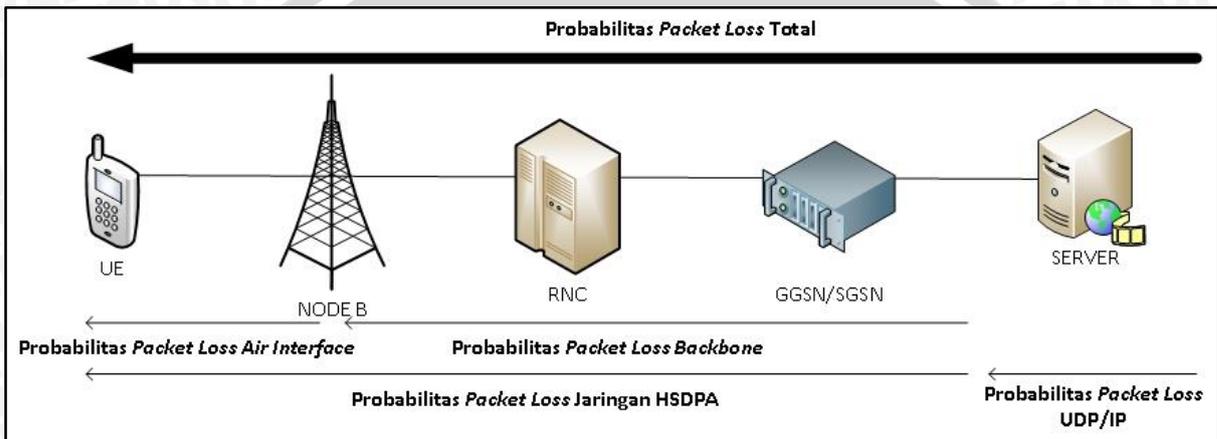
Delay end-to-end pada penelitian dapat ditentukan dengan persamaan 2.72.

$$t_{total} = t_{delay \text{ codec}} + t_{delay \text{ end-to-end HSDPA}} \quad (2.72)$$

Keterangan:

t_{total} = Delay total sistem (s)

2.4.3.3 Probabilitas Packet Loss pada HSDPA



Gambar 2.30 Probabilitas Packet Loss HSDPA

Sumber: Kumar Anurag, 2008

Probabilitas *packet loss* merupakan besarnya kemungkinan jumlah paket data yang hilang pada saat transmisi. Probabilitas *packet loss* total pada HSDPA ditentukan berdasarkan probabilitas *packet loss* pada jaringan tersebut serta pada *transport layer* UDP dan *network layer* seperti pada persamaan 2.73 (Kumar Anurag, 2008).

$$\rho_{tot} = 1 - [(1 - \rho_{net})(1 - \rho_{TCP/IP})] \quad (2.73)$$

dengan:

ρ_{tot} = Probabilitas *packet loss* total

ρ_{net} = Probabilitas *packet loss* pada jaringan HSDPA

$\rho_{TCP/IP}$ = Probabilitas *packet loss* model layer TCP/IP

Besarnya probabilitas *packet loss* dalam persen dirumuskan dalam persamaan 2.74.

$$\rho_{tot}(\%) = \rho_{tot} \times 100 \% \quad (2.74)$$

a) Probabilitas Packet Loss TCP/IP

Protokol UDP tidak memberikan jaminan bahwa paket akan dikirimkan semua sesuai dengan permintaan sampai tujuan (*connectionless*). *Packet loss* yang terjadi di *server* pada

transport layer dan *network layer* untuk protokol TCP maupun UDP pada model layer TCP/IP berdasarkan standar dari *White Paper* Qualcomm memiliki nilai, yaitu 10^{-8} .

b) Probabilitas *Packet Loss* pada Jaringan

Probabilitas *packet loss* pada jaringan HSDPA dihitung dari probabilitas *packet loss* yang terjadi pada *air interface* sistem HSDPA serta pada jaringan IP *backbone* yang meliputi RAN (*Node B* dan RNC) dan *Core Network* (SGSN dan GGSN). Probabilitas *packet loss* pada jaringan HSDPA dirumuskan dalam persamaan 2.75 hingga 2.80 (Kumar Anurag, 2008).

$$\rho_{\text{net}} = 1 - [(1 - \rho_{\text{AI}})(1 - \rho_{\text{IP backbone}})^h] \quad (2.75)$$

Keterangan:

- ρ_{net} = Probabilitas *packet loss* pada jaringan HSDPA
 ρ_{AI} = Probabilitas *packet loss* pada *air interface*
 $\rho_{\text{IP backbone}}$ = Probabilitas *packet loss* pada IP *backbone*
 h = Jumlah *hop* pada jaringan IP *backbone* (*Node B*, RNC, SGSN, GGSN)

1) Probabilitas *Packet Loss* pada *Air Interface*

Probabilitas *packet loss* pada *air interface* HSDPA ditentukan perhitungan meliputi perhitungan daya yang diterima kanal trafik, E_b/N_0 kanal trafik, dan probabilitas *error* kanal trafik sesuai dengan persamaan 2.76 hingga 2.79 (Vijay K. Garg, 2007). Nilai EIRP, *gain* antena *ms*, *Interference margin*, *bit rate ms*, *thermal noise* pada perhitungan disesuaikan dengan *downlink budget* HSDPA operator yang digunakan.

a. Daya yang Diterima Kanal Trafik

$$P_{\text{tr}} = \text{EIRP} - L + G_{\text{MS}} - I_{\text{m}} \quad (2.76)$$

dengan:

- P_{tr} = Daya yang diterima kanal trafik (dBm)
 EIRP = EIRP kanal trafik (dBm)
 L = *Pathloss* (dB)
 G_{MS} = *Gain* antena *mobile station* (dB)
 I_{m} = *Interference margin* (dB)

b. E_b/N_0 Kanal Trafik

$$E_b/N_0 = P_{\text{tr}} - 10 \log R_b - 10 \log 10^{0,1N_0} \quad (2.77)$$

Keterangan:

- E_b/N_0 = *Energy per bit-to-noise density ratio* kanal trafik (dB)
 P_{tr} = Daya yang diterima kanal trafik (dBm)
 R_b = *Bit rate* data pada *mobile station* (bps)

N_o = Thermal noise (dBm/Hz)

c. Probabilitas Error Kanal Trafik

$$P_e = \frac{e^{-E_b/N_o}}{\sqrt{\pi(E_b/N_o)}} \quad (2.78)$$

Keterangan:

P_e = Probabilitas *Bit Error Rate* (BER)

E_b/N_o = *Energy per bit-to-noise density ratio* (dB)

d. Probabilitas Packet Loss Air Interface HSDPA

$$\rho_{AI} = 1 - (1 - P_e)^{l+l'} \quad (2.79)$$

Keterangan:

ρ_{AI} = Probabilitas *packet loss air interface* HSDPA

P_e = Probabilitas *Bit Error Rate* (BER)

l = Panjang paket data (bit)

l' = Panjang *header* (bit)

2) Probabilitas Packet Loss pada IP Backbone

$$\rho_{IP \text{ backbone}} = W_{frame} \times P_e \quad (2.80)$$

Keterangan:

W_{frame} = Panjang *frame* tiap *node* (bit)

P_e = Probabilitas *Bit Error Rate* (BER) kanal trafik

2.4.3.4 Throughput

Throughput adalah banyaknya jumlah data yang diterima benar oleh penerima dari pengirim per satuan waktu. *Throughput* didapat dengan memperhatikan probabilitas *packet loss* (ρ) dan dihitung sesuai persamaan 2.81. (Mischa Schwartz, 1987)

$$\gamma = \frac{1}{t_v} = \frac{(1-\rho_{tot})}{t_1[1+(\alpha-1)\rho_{tot}]} \quad (2.81)$$

Keterangan:

γ = *Throughput* (paket/s)

t_v = Waktu rata-rata transmisi untuk mengirimkan paket yang benar (s)

t_1 = Waktu transmisi sebuah paket data atau *frame* (s)

ρ_{tot} = Probabilitas *packet loss* total

α = Konstanta perbandingan

Parameter α dihitung dengan menggunakan persamaan 2.82 (Mischa Schwartz, 1987).

$$\alpha = 3 + \frac{2t_p}{t_1} \quad (2.82)$$

Waktu total yang dibutuhkan untuk melewati sebuah paket dirumuskan dalam persamaan 2.83 (Mischa Schwartz, 1987).

$$t_p = \frac{t_{ptot}}{N_{paket}} \quad (2.83)$$

Waktu transmisi *frame* ditentukan dengan persamaan 2.84 (Mischa Schwartz, 1987).

$$t_1 = \frac{(PL_{frame} + H_{frame}) \times 8}{C_{trans}} \quad (2.84)$$

Keterangan:

- α = Konstanta perbandingan
- t_p = Waktu total yang dibutuhkan untuk melewati sebuah paket (s)
- t_{ptotal} = *Delay* propagasi total (s)
- t_1 = Waktu transmisi sebuah *frame* (s)
- N_{paket} = Jumlah paket yang ditransmisikan dari sumber sampai ke tujuan
- PL_{frame} = *Payload frame* (byte)
- H_{frame} = *Header frame* (byte)
- C_{trans} = Kapasitas saluran transmisi (bps)

2.5 Rekomendasi dan Standar Perangkat

2.5.1 Rekomendasi ITU-T G.1010

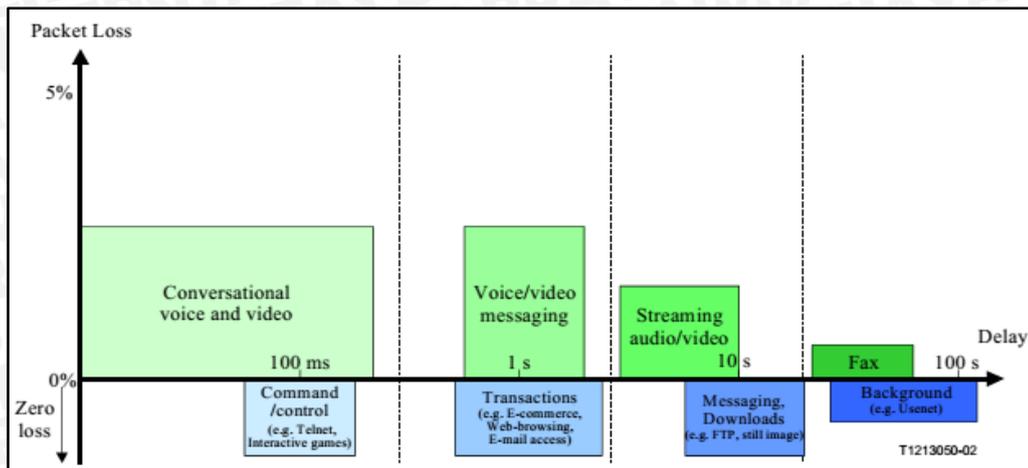
Berdasarkan ITU-T G.1010, Aplikasi memiliki persyaratan *packet loss* yang berbeda-beda. Tabel 2.7 menunjukkan syarat *packet loss* berdasarkan ITU-T G.1010.

Tabel 2.7 Standar *Packet Loss* ITU-T G.1010 untuk Aplikasi *Streaming*

| <i>Medium</i> | <i>Application</i> | <i>Degree of Symmetry</i> | <i>Information Loss</i> |
|---------------|-------------------------------------|---------------------------|-------------------------------------|
| <i>Audio</i> | <i>Conversational Voice</i> | <i>Two-way</i> | < 3% <i>Packet Loss Ratio (PLR)</i> |
| <i>Audio</i> | <i>Voice Messaging</i> | <i>One-way</i> | < 3% <i>PLR</i> |
| <i>Audio</i> | <i>High Quality Audio Streaming</i> | <i>One-way</i> | $\leq 1\%$ <i>PLR</i> |
| <i>Video</i> | <i>Videophone</i> | <i>Two-way</i> | < 1% <i>PLR</i> |
| <i>Video</i> | <i>Streaming</i> | <i>One-way</i> | $\leq 1\%$ <i>PLR</i> |

Sumber: ITU-T G.1010, 2002

Persyaratan untuk QoS parameter *delay* dan *packet loss* juga ditunjukkan pada Gambar 2.29.



Gambar 2.31 Mapping of User-Centric QoS Requirements
 Sumber: ITU-T, 2001

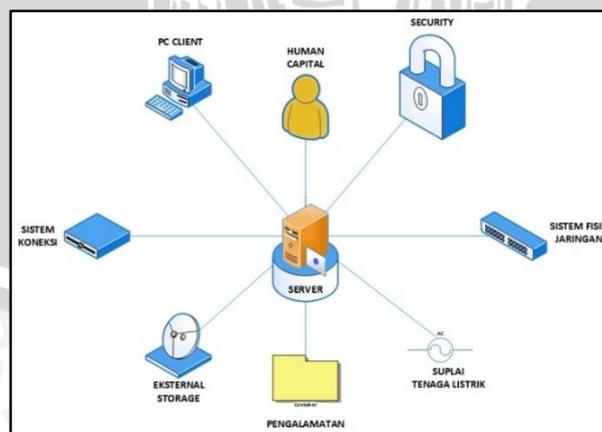
2.5.2 Perangkat Keras

Perangkat keras yang dibahas dalam penelitian adalah perangkat keras yang berada pada sisi penyedia layanan *video on demand*, dan pengguna. Untuk media atau jaringan yang berada di antara penyedia layanan dengan pengguna adalah milik operator telekomunikasi seluler di Indonesia.

Pada penelitian ini diperlukan perangkat keras utama, antara lain: PC server, *Network Interface Card* (NIC), Router, Modem, *Personal Computer* (PC) / Laptop, *Kabel Unshielded Twisted Pair* (UTP) dan konektornya, UPS serta Monitor.

a. PC Server

Server adalah terminal yang mengkoordinasikan semua aktivitas yang terjadi pada suatu sistem infrastruktur (sumber) dan dipengaruhi oleh beberapa faktor seperti yang ditunjukkan pada Gambar 2.32.



Gambar 2.32 Server dan Perangkatnya
 Sumber: opensource.telkomspeedy.com/wiki, 2010

Server dalam jaringan mempunyai tugas, yakni mengatur atau menjadi koordinator dalam jaringan, melakukan proses penyimpanan, pengolahan, pendistribusian data



secara terpusat. Server berbeda dengan varian komputer lainnya dan hal ini ditunjukkan oleh Tabel 2.8.

Tabel 2.8 Pembagian Komputer dan Segmentasinya

| Perangkat | Definisi | Ciri Khas | Pengguna |
|-----------------------------|---|--|---|
| Server | Komputer “induk” dari suatu sistem jaringan komputer (<i>network</i>) | Performa Tinggi. Beroperasi 24 jam x 7 hari <i>non-stop</i> . Harga relatif mahal. | Institusi, Bisnis, Kantor dan sebagainya. |
| PC Desktop | Komputer umum yang kita gunakan sehari-hari di mana saja umumnya diletakkan di meja | Beroperasi rata-rata maks. 18 jam/hari. Umumnya diletakkan di atas meja (<i>Desktop</i>) | Individu di rumah maupun di kantor |
| Notebook | Komputer “ <i>mobile</i> ” yang ditujukan untuk user yang “ <i>mobile</i> ” | Mudah dibawa, ringan dan memiliki performa mirip dengan <i>Desktop</i> . | <i>Mobile User</i> |
| <i>Consumer Electronics</i> | Komputer dengan ukuran kecil untuk aplikasi tertentu | Ringan, mudah dibawa kemana-mana. Hanya berfungsi tunggal. | MP3 <i>Player</i> , <i>Mobile Phone</i> , PDA |

Sumber: opensource.telkomspeedy.com/wiki, 2010

Untuk memenuhi kebutuhan dalam operasinya, server memiliki spesifikasi yang berbeda dengan pc pada umumnya. Untuk lebih jelasnya ditampilkan dalam Tabel 2.9.

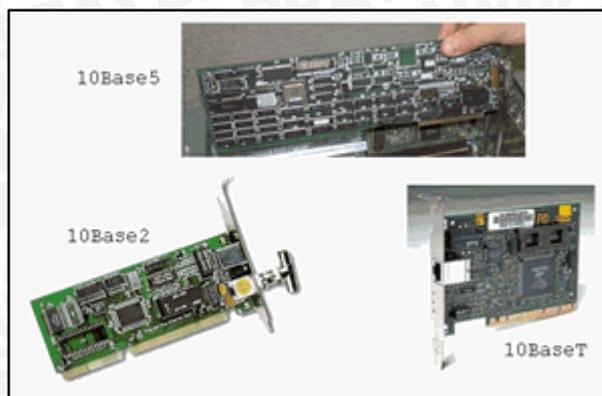
Tabel 2.9 Perbandingan *Device PC Desktop* dan *Server*

| Spesifikasi | PC Desktop | Server |
|------------------------|---|---|
| <i>Processor Intel</i> | Celeron D, Pentium 4, Core 2 Duo, Core 2 Quad | Xeon, Itanium2 |
| <i>Motherboard</i> | Intel 945G/P, 965G/P, G31, Q35, X38 | Intel 5000P/V/X, 3000, 3210 |
| RAM | <i>Unbuffered</i> (DDR, DDR2, DDR3) | ECC atau <i>ECC Registered</i> (DDR, DDR2) |
| | | <i>Fully Buffered</i> DIMM |
| HDD | ATA (<i>Parallel ATA</i>) | <i>Small Computer System Interface</i> (SCSI) |
| | SATA (<i>Serial ATA</i>) | SAS (<i>Serial Attached SCSI</i>) |
| <i>Video Graphics</i> | <i>Integrated min. 64 MB memory</i> | <i>Integrated only with 8/16MB memory</i> |
| <i>Power Supply</i> | 300-500 <i>watts</i> | 600-1000 <i>watts</i> |

Sumber: opensource.telkomspeedy.com/wiki, 2010

b. *Network Interface Card* (NIC)

NIC atau *Ethernet Card* adalah perangkat keras jaringan komputer sebagai adaptor pertukaran file. Fungsi perangkat ini adalah membantu pertukaran *file* dan data melalui jaringan komputer, membantu komputer-komputer untuk saling berkomunikasi dengan bantuan akses fisik media jaringan. NIC memiliki bentuk fisik seperti yang ditunjukkan Gambar 2.33.



Gambar 2.33 NIC

Sumber: jaringankomputer.org, 2011

NIC yang digunakan pada penelitian menggunakan soket *twisted pair*. Kecepatan operasional NIC tergantung pada standar protokol yang mendukung. Berikut adalah klasifikasi jenis *Ethernet*, antara lain:

- 10 Mbit/detik, diistilahkan *Ethernet* (standar yang digunakan adalah: 10Base2, 10Base5, 10BaseT, 10BaseF). Spesifikasi IEEE 802.3
- 100 Mbit/detik, atau disebut *Fast Ethernet* (standar yang digunakan: 100BaseFX, 100BaseT, 100BaseT4, 100BaseTX). Spesifikasi IEEE 802.3 3u
- 1000 Mbit/detik atau 1 Gbit/detik, disebut dengan *Gigabit Ethernet* (standar yang digunakan 1000BaseCX, 1000BaseLX, 1000BaseSX, 1000BaseT). Spesifikasi IEEE 802.3z
- 10000 Mbit/detik atau 10 Gbit/detik (opensource.telkomspeedy.com/wiki, 2010).

c. Router

Router adalah perangkat dalam jaringan untuk fungsi *routing* dan *forwarding*. Router menghubungkan dua atau lebih subnet.

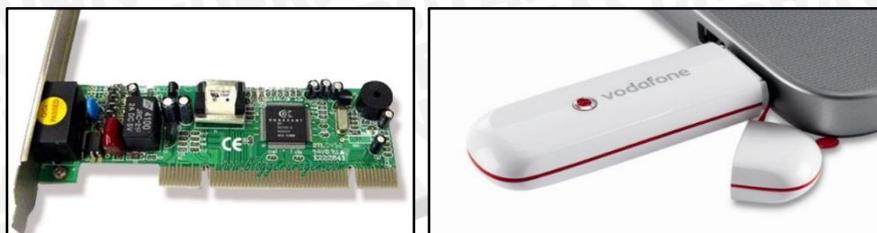
Router beroperasi di dua bidang yang berbeda:

- Panel Kontrol, dimana router belajar antarmuka yang paling tepat untuk meneruskan paket-paket khusus untuk tujuan tertentu.
- Panel *Forwarding*, yang bertanggung jawab untuk proses sebenarnya mengirimkan paket yang dari antarmuka umum menuju khusus.

d. Modem

Modem atau *Modulator Demodulator*. *Modulator* mengirimkan sinyal informasi ke dalam sinyal pembawa untuk dikirimkan, sedangkan *Demodulator* memisahkan sinyal informasi dari sinyal pembawa yang diterima sehingga informasi dapat dibaca. Modem adalah penggabungan Modulator dan Demodulator. Modem sampai saat ini dapat digunakan pada jaringan 2G hingga 3,75 G. Dari jenis penggunaannya, modem dapat

digolongkan menjadi modem internal (modem berada dalam *motherboard*) dan modem eksternal (modem menggunakan *interface* USB).



kiri ke kanan: Gambar 2.34 Modem Internal dan Eksternal
Sumber: tmcnet.com, 2014

e. Kabel *Unshielded Twisted Pair* (UTP)

Unshielded Twisted Pair (disingkat UTP) adalah kabel jaringan yang menggunakan bahan dasar tembaga, yang tidak dilengkapi dengan *shield* internal. UTP merupakan jenis kabel umum yang sering digunakan di dalam jaringan lokal (LAN), karena harganya yang rendah, fleksibel dan kinerja yang ditunjukkannya relatif bagus. Dalam kabel UTP, terdapat insulasi satu lapis yang melindungi kabel dari ketegangan fisik atau kerusakan tapi, tidak seperti kabel *Shielded Twisted-pair* (STP). Insulasi tersebut tidak melindungi kabel dari interferensi elektromagnetik.

Kemampuan transmisi data yang dimiliki kabel UTP tertulis dalam Tabel 2.10.

Tabel 2.10 Kategori UTP

| Kategori | Kegunaan |
|------------------------------------|--|
| <i>Category 1</i> (Cat1) | Kualitas suara analog |
| <i>Category 2</i> (Cat2) | Transmisi suara digital hingga 4 Mbps |
| <i>Category 3</i> (Cat3) | Transmisi data digital hingga 10 Mbps |
| <i>Category 4</i> (Cat4) | Transmisi data digital hingga 16 Mbps |
| <i>Category 5</i> (Cat5) | Transmisi data digital hingga 100 Mbps |
| <i>Enhanced Category 5</i> (Cat5e) | Transmisi data digital hingga 250 Mbps |

Sumber: opensource.telkomspeedy.com/wiki, 2010

2.5.3 Perangkat Lunak

2.5.3.1 Wireshark

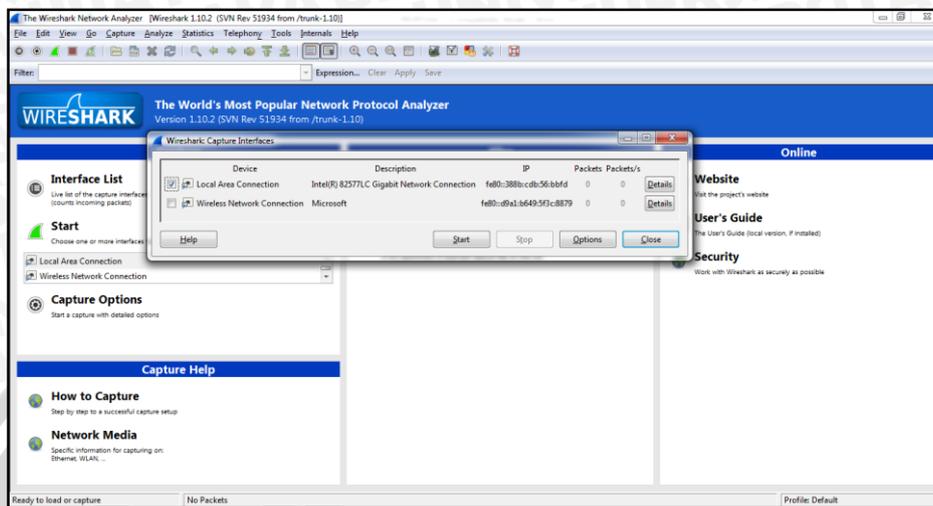
Wireshark adalah perangkat lunak *open source* dan bersifat gratis yang digunakan untuk menghitung, menganalisis paket data yang melewati suatu perangkat, komputer atau laptop.

Wireshark memiliki fungsi sebagai *packet sniffer*, yang memiliki 3 proses, antara lain:

- *Capture*. Wireshark menangkap paket data yang melewati adapter kabel jaringan maupun *wireless* pada pengguna.
- *Decode*. *Decode* atau dekode adalah proses mengubah suatu informasi pada protokol tertentu menjadi sebuah informasi bentuk lain yang diinginkan.

- *Analyze*. *Analyze* menampilkan beberapa parameter terkait performansi maupun karakteristik paket data.

Gambar 2.36 menunjukkan tampilan dari Wireshark meliputi *menubar*, *toolbar* dan *active sheet*.



Gambar 2.35 Lembar Kerja Wireshark

2.5.3.2 CentOS

CentOS adalah sistem operasi yang berbasis Linux yang dikembangkan oleh Red Hat Enterprise Linux (RHEL). CentOS bersifat *open-source* dan *freeware*.

Alasan untuk menggunakan sistem operasi CentOS, antara lain:

- CentOS Linux bersifat gratis dan bebas untuk didistribusikan ulang.
- Setiap sistem operasi CentOS Linux dapat digunakan hingga 10 tahun (dengan cara selalu *update* keamanan).
- Versi CentOS selalu terbaru setiap 2 tahun sekali.
- Karena sifatnya yang selalu *update* tiap 2 tahun maka CentOS memiliki keunggulan, yakni aman, sedikit pemeliharaan, dan stabil.

Kerja PC sebagai server akan membutuhkan sebuah sistem operasi yang lebih daripada sistem operasi untuk personal pada umumnya. Server harus mampu bekerja keras dan cepat serta stabil. Server untuk *multimedia streaming* secara dasar melakukan beberapa proses, yakni:

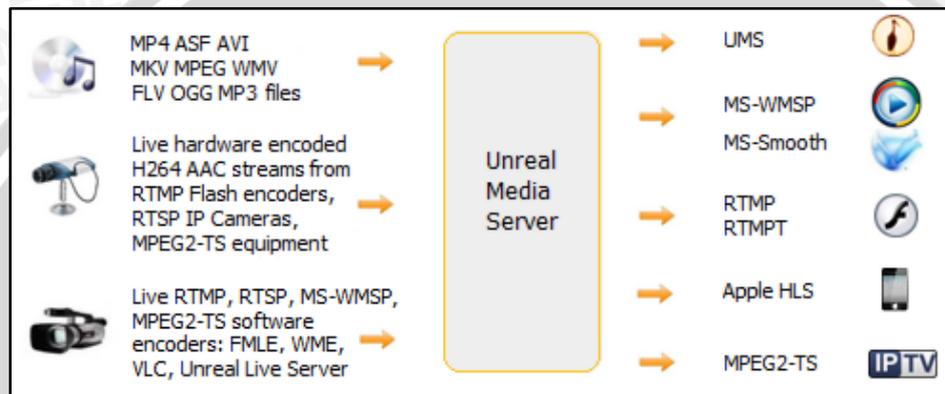
- *Virtual hosting*,
- *Webmail* dan *mail*
- *Media server*
- *Proxy* dan *DNS server*,

- *Firewall/router/gateway*,
- *Spam protection* dan lain-lain.

Penggunaan CentOS dibandingkan Windows Server terletak pada beberapa poin, yakni kecepatan *setup* untuk beberapa proses, kontrol *bandwidth*, *reverse proxy*, antivirus gratis dan proteksi *spam*. (centos.org/forums, 2007)

2.5.3.3 Unreal Media Server dan Unreal Media Player

Unreal Media Server adalah perangkat lunak untuk media *streaming server live* maupun *on-demand streaming*. Unreal menggunakan berbagai macam protokol (H.264 AVC, MPEG-2, MKV dan lain-lain).



Gambar 2.36 Unreal Media Server
Sumber: umediaserver.net, 2014

Berdasarkan Gambar 2.37, Unreal Media Server mampu menyiarkan format multimedia. Format-format multimedia yang didukung antara lain: MP4, ASF, AVI, MPEG, WMV, FLV, OGG, MP3, 3GP, dan MOV (umediaserver.net, 2014).

2.5.4 Standar Protokol dan Perangkat

2.5.4.1 Header Protokol dan Frame

Header adalah bit-bit yang berisi informasi tujuan paket data dikirimkan, urutan maupun sumber paket data itu sendiri. *Header* memiliki panjang dengan satuan *octet* atau byte. Panjang *header-header* protokol yang digunakan pada penelitian ini ditunjukkan oleh Tabel 2.11.

Tabel 2.11 Header Format

| Protokol | Kuantitas | Sumber |
|----------------------|-----------|---|
| RTP Header | 12 byte | Peterson, L, et al. 2007. <i>Computer Networks</i> . San Fransisco: Morgan Kaufmann: 430 |
| IPv4 Header | 20 byte | RFC 791 |
| UDP Header | 8 byte | Stalling Williams. 2007. <i>Data and Computer Communications</i> . USA: Prentice Hall: 39 |
| MTU Ethernet | 1500 byte | RFC 1191 |
| GTP header | 8 byte | ETSI TS 129 060 V5.14.0 |
| Frame Check Sequence | 4 byte | IEEE 802.3 |

| | | |
|----------------------------|---------|--|
| Ethernet Header | 14 byte | IEEE 802.3 |
| CPCS PDU/AAL5 Header | 8 byte | ITU-T I.363.3 |
| AAL5 Header | 8 byte | RFC 2684 |
| PDCP Header | 1 byte | ETSI TS 125 323 V3.6.0 |
| RLC PDU size | 40 byte | ETSI TS 125 322 V6.3.0, Dahlman, Erik et al. 2008. <i>3G Evolution HSPA and LTE for Mobile broadband</i> . Oxford: AP |
| RLC Header | 2 byte | ETSI TS 125 322 V6.3.0, Dahlman, Erik et al. 2008. <i>3G Evolution HSPA and LTE for Mobile broadband</i> . Oxford: AP |
| ATM Header | 5 byte | RFC 2684 |
| ATM payload | 48 byte | RFC 2684 |
| Transport Block Size (TBS) | 7298 | Siemens HSDPA White Paper |
| MAC-d Header | 4 bit | Dahlman, Erik et al. 2008. <i>3G Evolution HSPA and LTE for Mobile broadband</i> . Oxford: AP |
| MAC-d SDU | 42 byte | Dahlman, Erik et al. 2008. <i>3G Evolution HSPA and LTE for Mobile broadband</i> . Oxford: AP |
| MAC-hs Header | 21 bit | Johnson, Chris. 2008. <i>Radio Access Networks for UMTS: Principles and Practice</i> . England: John Wiley & Sons Ltd. |
| AAL2 Header | 3 byte | 3GPP TSG-RAN Working Group 3 IP UTRAN |
| HS-DSCH FP Header | 7 byte | 3GPP TS 25.425 |
| CRC Header | 2 byte | RACOM, 2014 |
| Header PDCP | 1 byte | 3GPP TS. 25. 323 Rel.5 |

2.5.4.2 Standar Kecepatan Perangkat

Perangkat keras baik pada sisi server, jaringan operator maupun pengguna memiliki standar terkait pemrosesan data maupun *interface* perangkat. Standar kecepatan perangkat untuk pemrosesan maupun *interface* untuk penelitian ini ditunjukkan oleh Tabel 2.12.

Tabel 2.12 Standar Perangkat

| Standar | Kuantitas | Sumber |
|--------------------------|---|----------------------------|
| Server Interface | 10Gbps over ethernet | HP Proliant Server |
| SGSN bitrate | With GGSN 2.4 Gbps (3G) 380 Mbps (2G) With RNC STM-1, 155,52 Mbps | Alcatel Lucent 7500 |
| GGSN bitrate interface | 25 Gbps | Alcatel Lucent 7750 |
| RNC bitrate interface | STM-1, 155,52 Mbps | Erricson 3810 review vol.2 |
| Node B bitrate interface | STM-1, 155,52 Mbps | Erricson 3810 review vol.2 |

2.5.4.3 HSDPA Terminal Categories

HSDPA *Terminal Categories* adalah kategori yang menunjukkan ukuran kecepatan pengolahan data, modulasi dan lain-lain pada *User Equipment* (UE) jaringan HSDPA. Kategori ini berguna dalam perhitungan *delay* maupun parameter lain pada UE dan ditunjukkan Tabel 2.13.

Tabel 2.13 HSDPA *Terminal Capability Categories*

| Category | Mod. Scheme | Max. Number of parallel codes HS-DSCH | Min. inter-TTI interval | Transport channel bits per TTI | ARQ type at maximum data rate | Achievable max. Data rate (Mbps) |
|----------|-------------|---------------------------------------|-------------------------|--------------------------------|-------------------------------|----------------------------------|
| 1 | 16QAM | 5 | 3 | 7298 | Soft | 1,2 |
| 2 | 16QAM | 5 | 3 | 7298 | IR | 1,2 |
| 3 | 16QAM | 5 | 2 | 7298 | Soft | 1,8 |
| 4 | 16QAM | 5 | 2 | 7298 | IR | 1,8 |
| 5 | 16QAM | 5 | 1 | 7298 | Soft | 3,6 |
| 6 | 16QAM | 5 | 1 | 7298 | IR | 3,6 |
| 7 | 16QAM | 10 | 1 | 14411 | Soft | 7,2 |
| 8 | 16QAM | 10 | 1 | 14411 | IR | 7,2 |
| 9 | 16QAM | 15 | 1 | 20251 | Soft | 10,2 |
| 10 | 16QAM | 15 | 1 | 27952 | IR | 14,4 |
| 11 | QPSK | 5 | 2 | 3630 | Soft | 0,9 |
| 12 | QPSK | 5 | 1 | 3630 | Soft | 1,8 |

Sumber: Harri Holma & Antti Toskala, 2007

Category menunjukkan jenis-jenis kategori perangkat *terminal user* atau *User Equipment* (UE). Ada 12 kategori UE dengan jenis modulasi yang berbeda. *Modulation Scheme* (disingkat *Mod. Scheme*) adalah tipe-tipe modulasi yang digunakan dari *Node B* menuju UE. Modulasi yang digunakan adalah modulasi digital, 16 QAM dan QPSK. Penggunaan kedua modulasi ini didasarkan keadaan kanal. *HS-DSCH codes received* adalah banyaknya kode maksimal dari *Node B* yang dikirimkan menuju UE. Kode ini membedakan kanal-kanal yang masing-masing membawa informasi berbeda-beda. Satu UE bisa menerima 5, 10 atau 15 kode tergantung kemampuan UE. 15 berasal banyaknya jenis kode dari modulasi 16 QAM dengan maksimal pengkodean untuk 16 kanal. *Minimum inter-TTI interval* menunjukkan rentang waktu minimal untuk TTI. *Transport channel bits per TTI* menunjukkan besarnya *payload* TBS untuk setiap TTI. *ARQ type at maximum data rate* menunjukkan tipe HARQ, *soft combining* atau *Incremental Redundancy* (IR). *Soft combining* menunjukkan tipe transmisi yang menggabungkan antara *systematic bits* dengan *parity bits*, sedangkan IR menggunakan tipe transmisi yang membedakan antara *systematic bits* dengan *parity bits*. *Achievable data rate* menunjukkan besar *data rate* yang bisa diterima terminal (UE) secara teoretis (Harri Holma & Antti Toskala, 2007).