

## BAB II

### TINJAUAN PUSTAKA

Perkembangan telekomunikasi di dunia internasional, termasuk Indonesia, berjalan sangat pesat. Perkembangan ini dimulai dari teknologi AMPS (*Advanced Mobile Phone System*) yang hanya mengandalkan layanan *voice* dengan menggunakan teknologi analog, dimana tidak boleh ada dua pengguna yang menggunakan kanal yang sama baik dalam satu sel maupun sel tetangganya. Teknologi AMPS yang memiliki banyak kelemahan digantikan oleh teknologi lain yaitu GSM (*Global System for Mobile Communication*) yang mampu menyalurkan komunikasi suara dan data berkecepatan rendah, yang kemudian berkembang menjadi GPRS (*General Packet Radio Service*) dengan layanan *Packet Switched*. Pada fase berikutnya, diperkenalkan EDGE (*Enhanced Data rates for GSM Evolution*) dengan layanan *Packet Switched* dan kemampuan telekomunikasi yang lebih fleksibel, *real time* dan bahkan memberikan layanan yang lebih mengeksplorasi kemampuan multimedia. Pada pengembangan selanjutnya, EDGE berkembang menjadi teknologi WCDMA (*Wideband Code Division Multiple Access*) yang memberikan kecepatan akses data yang tinggi. Selanjutnya WCDMA berkembang menjadi HSDPA (*High Speed Downlink Packet Access*) yang bertujuan mampu meningkatkan kemampuan performansi *downlink* jaringan WCDMA.

LTE (*Long Term Evolution*) atau yang disebut dengan teknologi 4G merupakan suatu teknologi telekomunikasi bergerak yang dikeluarkan oleh 3GPP Release 8, dan merupakan teknologi telekomunikasi seluler sebagai evolusi atau pengembangan dari teknologi UMTS/WCDMA/HSPA.

#### 2.1 *Universal Mobile Telecommunication System (UMTS)*

UMTS (*Universal Mobile Telecommunication System*) merupakan kelanjutan dari teknologi GSM/GPRS dimana perbedaan utamanya adalah kemampuan akses data yang lebih cepat. Aplikasi generasi ketiga tidak hanya menyediakan layanan baru, tapi juga memperbaiki layanan sebelumnya dengan kemampuan yang baru seperti *delay* yang lebih kecil dan layanan yang lebih bervariasi.

Sesuai dengan standart IMT 2000 (*International Mobile Telecommunication 2000*) yang merupakan standarisasi teknologi generasi ketiga, kecepatan data dari jaringan UMTS ini sebesar 144kbps/384kbps untuk cakupan area yang luas dengan

mobilitas tinggi, serta 2Mbps untuk cakupan local dengan mobilitas rendah. UMTS dapat digunakan untuk layanan hiburan komersil, pengiriman informasi dengan *bandwidth* yang besar untuk pengguna yang bergerak melalui jaringan umum dan satelit. (Sauter, 2011:118)

### 2.1.1 Arsitektur Jaringan *Universal Mobile Telecommunication System* (UMTS)

Secara sederhana arsitektur UMTS terdiri dari:

#### 1. Bagian peralatan pelanggan (*User Equipment*)

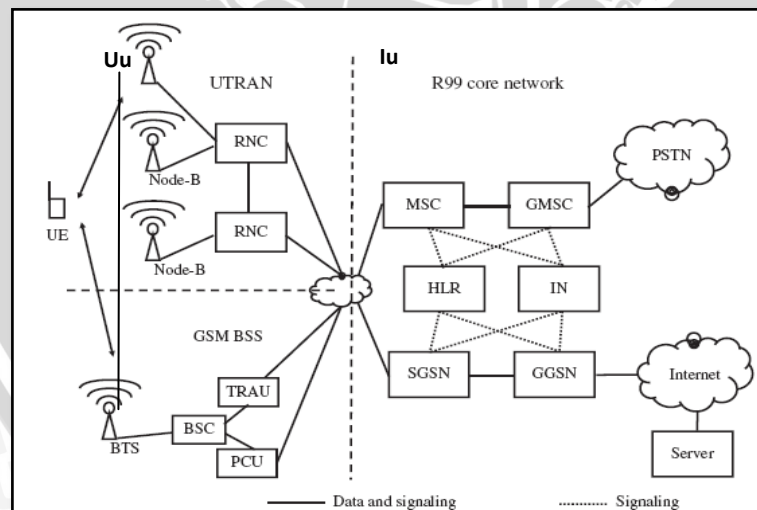
UE adalah perangkat yang digunakan untuk mengakses layanan UMTS. Bagian tersebut terdiri dari:

##### a) *Mobile Equipment* (ME)

Merupakan terminal radio yang digunakan untuk komunikasi radio dengan jaringan UMTS melalui *Uu Interface*.

##### b) *User Identity Module* (UIM)

Merupakan kartu yang menangani identitas pelanggan dan berisi informasi langganan yang dibutuhkan di terminal, termasuk melakukan pengesahan algoritma dan beberapa informasi layanan yang dibutuhkan di terminal.



**Gambar 2.1:** Arsitektur Jaringan UMTS  
(Sumber : Sauter, 2011:118)

#### 2. Bagian Infrastruktur

Bagian ini terdiri atas *node* yang mempunyai berbagai fungsi untuk memberikan layanan kepada pengguna dan juga memutuskan *interface*



radio. Infrastruktur adalah sumber yang dibagi (*shared resource*) untuk memberikan layanan pada semua pengguna yang berada dalam satu wilayah cakupan. Bagian dari infrastruktur terdiri dari:

a) Jaringan UTRAN *Access Network*

Blok fungsional dari UTRAN terbagi dua, yaitu *NodeB* dan *Radio Network Controller* (RNC). *NodeB* dapat dikatakan BTS, sedangkan RNC ekuivalen dengan BSC. RNC dan *NodeB* terhubung dengan *Iu Interface*.

Agar mampu melayani paket data maka UTRAN dilengkapi dengan elemen *hardware* dan *software* baru. UTRAN memerlukan perangkat *Packet Control Unit* (PCU) untuk menangani paket data. PCU bertanggung jawab terhadap *layer Radio Link Control* (RLC) dan *Medium Access Control* (MAC) melalui interface udara.

➤ *Radio Network Controller* (RNC)

RNC merupakan sentral BTS-BTS dalam UMTS yang disebut *NodeB*. Beberapa fungsi dari RNC adalah:

1) UMTS *Radio Resource Management*

*Radio Resource Management* berfungsi untuk mengatur konfigurasi dari kanal, trafik, kontrol kanal, *handover* parameter dan *power control*.

2) *Telecom Functionality*

Bagian ini berfungsi untuk manajemen lokasi dan koneksi ke *node* lain, manajemen kanal *Iu Interface* dan *Iub Interface*, ATM switching dan multiplexing.

3) *Network Operation and Maintenance*

Bagian ini berfungsi untuk mengawasi kontinuitas jaringan. Beberapa kemungkinan masalah yang ada dalam suatu jaringan pertama kali harus diketahui lokasi suatu kerusakan yang disebut *fault localization*. Lokasi kerusakanyang diketahui harus dianalisa apakah *software* perlu diperbaiki dengan menggunakan *database* sebelumnya atau penggantian perangkat rusak.

➤ *Node-B*

Node B digunakan untuk mentransmisikan data dari *lub-interface* dan *Uu-interface* atau sebaliknya, serta ikut mengontrol sumber radio. Satu *NodeB* dapat terdiri 1 sampai 3 cell dan setiap *Node B* dikontrol oleh satu RNC. *NodeB* terdiri atas transmisi, TRX, *duplexer*, *combiner*, *baseband unit*, antena dan *control processor unit*.

b) *Core Network*

*Core Network* terdiri dari beberapa bagian yaitu:

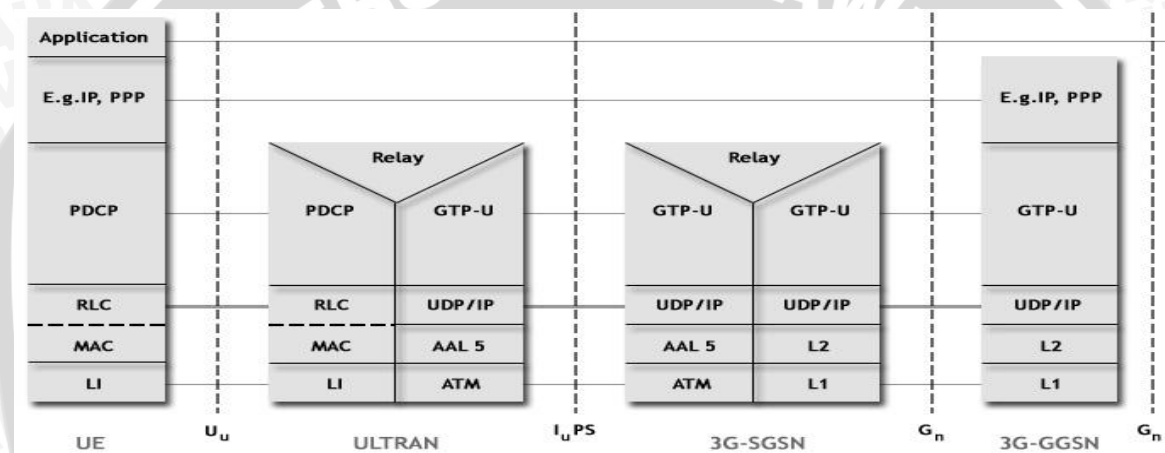
- SGSN (*Serving GPRS Support Node*) bertugas untuk mengarahkan perjalanan paket-paket data antara *Core Network* dengan jaringan akses UTRAN. SGSN dihubungkan dengan RNC dan melayani akses yang menuju jaringan paket untuk UE.
- GGSN (*Gateway GPRS Support Node*) bertugas sebagai *interface* antara jaringan *backbone Packet Switching* (PS) UMTS dengan jaringan eksternal PDN (*Packet Data Network*), seperti internet.
- CPS (*Call Processing Server*) atau MSC berfungsi untuk menangani *switching* dan *signaling* untuk MS yang berada di dalam area geografis layanan. Untuk menangani permintaan panggilan CPS dapat mengakses informasi dari HLR/HA dan (E/R)/FA serta *Authentication Center*.
- HLR/HA dan (E/R)/FA dijalankan pada jaringan yang mempunyai fungsi sama. Keduanya merupakan unit fungsional yang digunakan untuk registrasi pelanggan. HLR/HA adalah *home register* yang berisi data statis yaitu profil *user* jenis layanan dan layanan tambahan, serta data dinamis yang berisi identifikasi jaringan dan lokasi terbaru UE. FA sebagai register penunjang yang dapat dianggap sebagai data dinamik yang secara intensif bertukar data dengan HA.
- *Authentication Center* berfungsi untuk melayani aturan keamanan dan konfirmasi.



### 2.1.2 Arsitektur Protokol Stack Jaringan UMTS

Arsitektur Protokol UMTS untuk *User Plane* pada Gambar 2.2. UTRAN RNC memiliki *link* komunikasi *Asynchronous Transfer Mode* (ATM) dengan 3G-MSC dan SGSN. Dalam sebuah *link* ATM informasi yang akan ditransmisikan dibagi menjadi sel-sel berukuran tetap, *multiplexing* dan dipancarkan.

Arsitektur Protokol UMTS menggunakan *ATM Adaptation Layer 5* (AAL5). Protokol layer *adaption* menyediakan dukungan untuk data *bit rate* variabel *connection-oriented*, atau *connectionless* dan jasa. Terdapat lapisan *Packet Data Convergence Protocol* (PDCP) dan *GPRS Tunneling Protocol for the User Plane* (GTP-U).



**Gambar 2.2 :** *User Plane Protocol* Arsitektur UMTS

(Sumber : Anonymous, *Wireless Cellular Network Technologies: 3G Mobile Telephony UMTS*, 2009)

#### 1) *Radio Resource Control* (RRC)

RRC adalah bagian dari *control plane* yang bertanggung jawab untuk protokol PDCP, RLC, MAC, dan PHY layer 1 dan layer 2. Layanan dan fungsi utama dari RRC adalah *admission control*, manajemen *handover*, manajemen *Quality of Service* (QoS), mengatur dan melaporkan pengukuran stasiun terminal.

#### 2) *Packet Data Convergence Protocol* (PDCP)

Fungsi utama dari PDCP untuk *user plane* adalah mengompres *header IP*, pengiriman data dari pengguna dan *chipering*. Pada *control plane*, PDCP bertanggung jawab dalam fungsi pengendalian pengiriman data dan *ciphering*.

#### 3) *Radio Link Control* (RLC)

Segmentasi dan *reassembly* paket data dari layer yang lebih tinggi serta perbaikan kesalahan yang memberikan koreksi kesalahan dengan retransmisi dalam

modus transfer data adalah fungsi utama dari RLC. Selain itu, *flow control* antara eNodeB dan *mobile terminal* juga ditangani oleh RLC.

#### 4) *Medium Access Control* (MAC)

MAC bertanggung jawab untuk penjadwalan *uplink* dan *downlink*, perbaikan kesalahan, modulasi adaptif dan pemetaan antenna, serta sebagai pemantauan volume lalu lintas. MAC menerima RLC PDU bersama dengan informasi status pada jumlah data dalam transmisi RLC.

#### 5) UDP/IP

Unit data Protocol (UDP) dan IP digunakan sebagai standar dan transportasi dasar IP. UDP digunakan sebagai pengganti Transmission Control Protocol (TCP) karena lapisan yang lebih tinggi sudah menyediakan transportasi yang handal dengan pemulihan kesalahan.

#### 6) Layer Fisik (PHY)

Fungsi utama dari layer fisik adalah *coding*, modulasi dan transmisi *multiple* antenna. (Holma, Harri, Antti Toskala. 2007)

## 2.2 Long Term Evolution (LTE)

LTE disebut juga sebagai E-UTRA (*Evolved Universal Terrestrial Radio Access*) dan diperkenalkan oleh 3GPP sebagai *release* 8. LTE merupakan teknologi telekomunikasi seluler sebagai evolusi atau pengembangan dari teknologi UMTS/WCDMA/HSPA. LTE menggunakan spesifikasi *air interface* yang baru untuk meningkatkan kecepatan data dibandingkan dengan HSPA. Perbedaan utama antara LTE, WCDMA dan HSPA adalah penggunaan teknologi OFDMA pada sisi *downlink* dan SC-FDMA pada sisi *uplink*-nya. LTE mempunyai kemampuan mengirimkan data dengan kecepatan tinggi mencapai 100 Mbps untuk *downlink* dan 50 Mbps untuk *uplink*-nya. Peningkatan kecepatan ini juga disebabkan oleh kemampuan LTE untuk menggunakan teknologi MIMO (*Multi Input Multi Output*) dan *bandwidth*-nya yang *scalable* mulai dari 1,4 MHz sampai 20 MHz (K. Fazel and S. Kaiser, 2008)

### 2.2.1 User Equipment (UE)

UE merupakan perangkat atau terminal pada sisi pelanggan yang berupa *handset* untuk mengirim dan menerima informasi. LTE memiliki 5 tipe UE. Karakteristik dari 5



tipe berdasarkan *peak data rates* pada *down link* maupun *up link*, dan metode modulasi yang didukung, ditampilkan dalam Tabel 1.

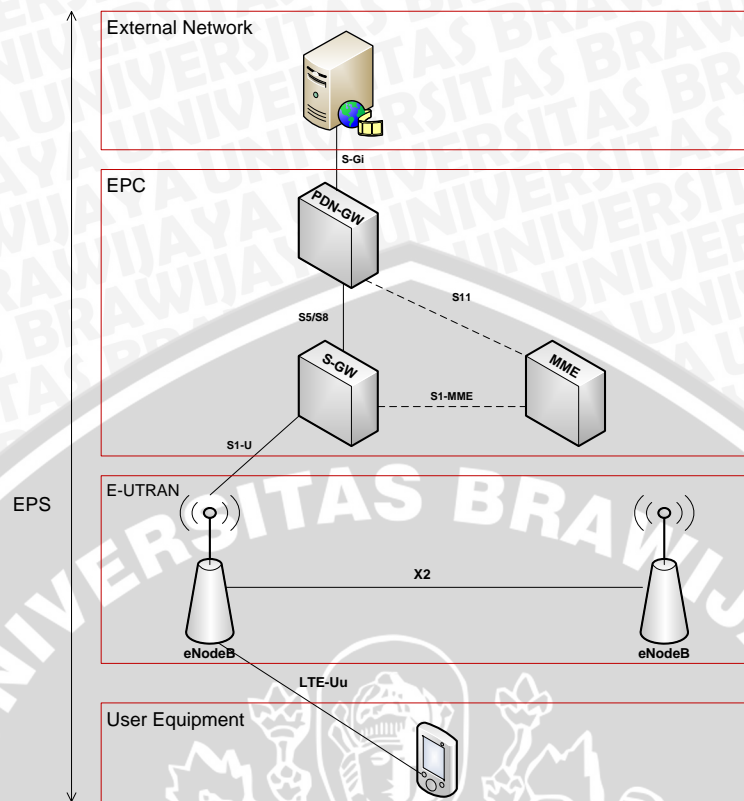
Tabel 2.1: *Category UE Pada LTE*

Category		1	2	3	4	5
Peak rate Mbps	DL	10	50	100	150	300
	UL	5	25	50	50	75
Capability for physical functionalities						
RF bandwidth		20MHz				
Modulation	DL	QPSK, 16QAM, 64QAM				
	UL	QPSK, 16QAM				QPSK, 16QAM, 64QAM
Multi-antenna						
2 Rx diversity		Assumed in performance requirements.				
2x2 MIMO		Not supported	Mandatory			
4x4 MIMO		Not supported				Mandatory

(Sumber : Gessner, 2008)

### 2.2.2 Evolved Packet System (EPS)

EPS *framework* terdiri dari *Evolved Packet Core* (EPC) dan *Evolved UMTS Terrestrial Radio Access Network* (E-UTRAN) seperti yang ditunjukkan pada Gambar 2.3. EPC berhubungan dengan E-UTRAN dan yang lainnya. EPC berisi *Mobile Management Entity* (MME), *System Architecture Evolution Gateway* (S-GW) dan *Packet Data Network Gateway* (PDN-GW). E-UTRAN hanya berisi *Evolved Universal Terrestrial Radio Access Network Base Stations* (eNodeB) dimana *User Equipment* (UE) berhubungan dengan eNB dan eNB berhubungan dengan EPC dan yang lainnya.



**Gambar 2.3.** Arsitektur LTE  
(Sumber : H. Holma dan A. Toskala, 2009)

a. PDN-GW (*Packet Data Network Gateway*)

PDN-GW merupakan jangkar untuk mobilitas antara 3GPP dengan teknologi *non-3GPP* seperti WiMAX, 3GPP2 dan WLAN (*Wireless Local Area Network*) melalui beberapa *interface*. PDN-GW menyediakan konektivitas dengan jaringan paket data eksternal. Selain itu PDN-GW bertanggung jawab untuk mengalokasikan alamat IP UE, pengisian aliran data berdasarkan aturan dari PCRF dan menyaring *downlink* paket IP *user* kedalam *bearer* QoS yang berbeda.

b. S-GW (*Serving Gateway*)

Semua pengguna paket IP dikirimkan melalui S-GW, yang berfungsi sebagai jangkar untuk mobilitas data *bearer* bila UE bergerak di antara eNodeB. S-GW juga bertanggung jawab dalam menetapkan *routing* dan meneruskan paket data *user* dengan *interface* S-1, menangani kompresi *header* IP serta enkripsi data *user*.

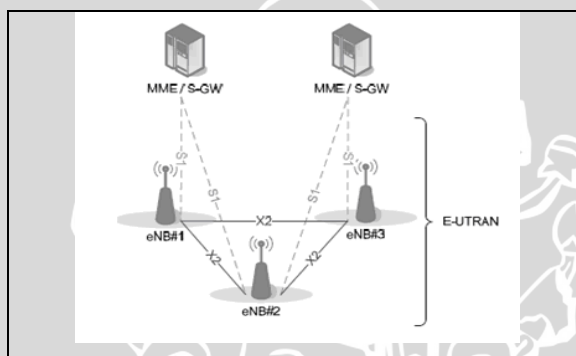


c. MME (*Mobility Management Entity*)

MME adalah *node* kontrol yang memproses sinyal antara UE dan CN (*Core Network*)/EPC. Selain itu MME juga berfungsi sebagai autentifikasi dan keamanan serta *mobility management*. Protokol yang berjalan antara UE dan CN dikenal sebagai protokol *Non-Access Stratum* (NAS). MME juga menyediakan fungsi *control plane* untuk mobilitas antara LTE dan jaringan akses 2G/3G. (Stefania Sesia, Issam Toufik dan Matthew Baker, 2009 : 25)

### 2.2.3 Evolved UMTS Terrestrial Radio Access Network (E-UTRAN)

Jaringan akses LTE, E-UTRAN, hanya terdiri dari eNodeB seperti yang ditunjukkan pada Gambar 2.4. Untuk trafik pengguna biasa, tidak ada pengontrol terpusat pada E-UTRAN, maka arsitektur E-UTRAN bisa dikatakan *flat*.

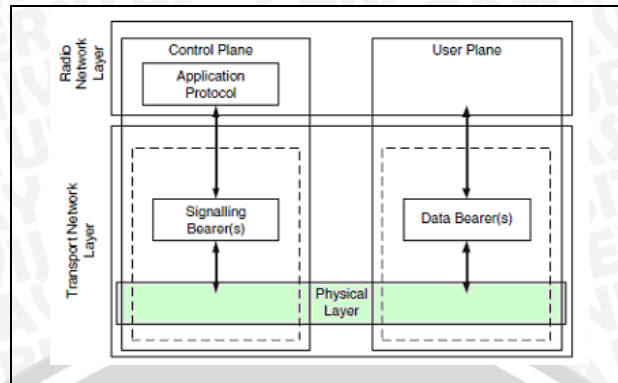


**Gambar 2.4** Arsitektur E-UTRAN

(Sumber : Stefania Sesia, Issam Toufik and Matthew Baker, 2009)

eNodeB biasanya saling terhubung satu sama lain melalui *interface* yang dikenal sebagai X2, dan pada EPC melalui *interface* yang lebih khusus yaitu S1, dengan MME melalui *interface* S1-MME dan S-GW melalui *interface* S1-U. Protokol yang bekerja antara eNodeB dan UE disebut protokol *Access Stratum* (AS). (Stefania Sesia, Issam Toufik and Matthew Baker, 2009)

eNB merupakan *entity* yang hanya ada didalam E-UTRAN yang merupakan *interface* dengan UE melalui *interface* LTE-UE. eNB merupakan *hosts* dari layer fisik, MAC, *Radio Link Control* (RLC) dan *Packet Data Control Protocol* (PDCP). eNB bisa mendukung *mode* FDD, TDD atau operasi *dual mode* dengan model protokol yang ditunjukkan pada Gambar 2.5. (Mustafa Ergen, 2009)



**Gambar 2.5** Model Protokol E-UTRAN

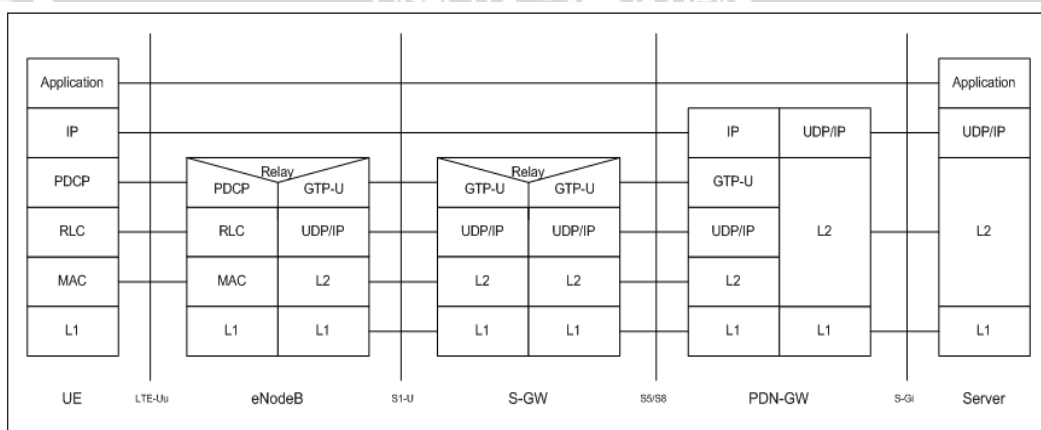
(Sumber : Mustafa Ergen, 2009)

## 2.2.4 Arsitektur Protokol Radio E-UTRAN

### 2.2.4.1 User Plane

Semua paket IP untuk UE dienkapsulasi dalam sebuah protokol khusus EPC dan di jalur antara PDN-GW dan eNodeB untuk transmisi ke UE. Protokol *tunneling* yang berbeda digunakan pada *interface* yang berbeda. Sebuah protokol *tunneling* khusus disebut *GPRS Tunneling Protocol (GTP)* yang digunakan melalui *interface core network*.

Protokol *stack user plane* E-UTRAN ditunjukkan pada Gambar 2.6, yang terdiri dari sublayer *Packet Data Convergence Protocol (PDCP)*, *Radio Link Control (RLC)* dan *Medium Access Control (MAC)*, yang diterminasi di eNodeB pada sisi jaringan. (Sumber : Stefania Sesia, Issam Toufik and Matthew Baker, 2009)



**Gambar 2.6** User Plane Protocol Stack E-UTRAN

(Sumber : Stefania Sesia, Issam Toufik and Matthew Baker, 2009)



1) *Radio Resource Control (RRC)*

RRC adalah bagian dari *control plane* yang bertanggung jawab untuk protokol PDCP, RLC, MAC, dan PHY layer 1 dan layer 2. Layanan dan fungsi utama dari RRC adalah *admission control*, manajemen *handover*, manajemen *Quality of Service (QoS)*, mengatur dan melaporkan pengukuran stasiun terminal.

2) *Packet Data Convergence Protocol (PDCP)*

Fungsi utama dari PDCP untuk *user plane* adalah mengompres *header IP*, pengiriman data dari pengguna dan *chipering*. Pada *control plane*, PDCP bertanggung jawab dalam fungsi pengendalian pengiriman data dan *ciphering*.

3) *Radio Link Control (RLC)*

Segmentasi dan *reassembly* paket data dari layer yang lebih tinggi serta perbaikan kesalahan melalui ARQ adalah fungsi utama dari RLC. Selain itu, *flow control* antara eNodeB dan *mobile terminal* juga ditangani oleh RLC.

4) *Medium Access Control (MAC)*

MAC bertanggung jawab untuk penjadwalan *uplink* dan *downlink*, perbaikan kesalahan, modulasi adaptif dan pemetaan antena.

5) Layer Fisik (PHY)

Fungsi utama dari layer fisik adalah *coding*, modulasi dan transmisi *multiple antena*. (K. Fazel dan S. Kaiser, 2008)

### 2.3 Video Streaming

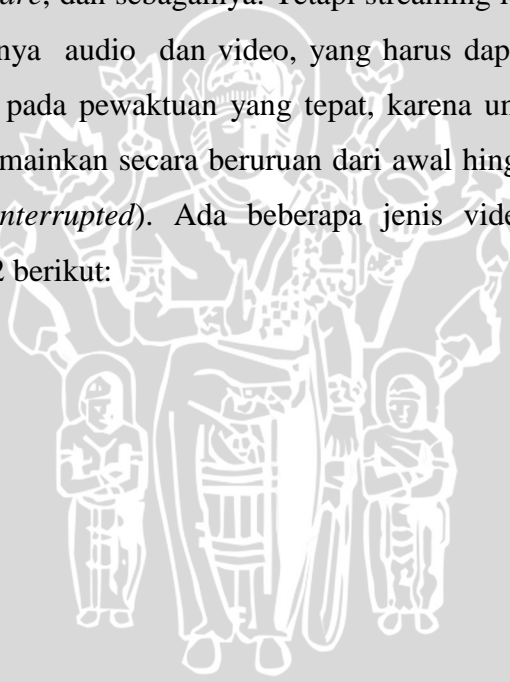
Awalnya telekomunikasi dua arah langsung yang semula hanya berupa audio saja, kini telah berkembang menjadi audio sekaligus video yang tercakup dalam dua arah percakapan jarak jauh. *Teleconference* dan *video streaming* internet menjadi contoh generasi terbaru sistem komunikasi global.

*Video streaming* merupakan salah satu teknologi telekomunikasi yang bersifat *real time* serta dapat menyalurkan informasi berupa audio maupun video dengan menggunakan jaringan *Internet Protocol (IP)*. Dengan teknologi *video streaming* ini, *user* tidak perlu menunggu hingga *file* selesai di-*download* secara keseluruhan untuk memainkannya. Sebaliknya, *user* dapat memainkan media dengan menunggu beberapa detik saja. Akan tetapi, ada beberapa permasalahan yang mempengaruhi performansi dari *video streaming* diantaranya adalah video berhenti berjalan atau bergerak lambat saat kita sedang menonton video sehingga kita harus menunggu beberapa waktu hingga

video kembali berjalan. Selain itu, kualitas video yang dihasilkan juga seringkali buruk dimana video *streaming* yang kita amati kurang jernih dan cenderung tidak jelas.

Dalam dunia internet, *streaming* lebih mengacu kepada teknologi yang mampu mengompresi atau menyusutkan ukuran *file* audio dan video agar mudah dikirimkan melalui jaringan internet. Pengiriman *file* audio dan video tersebut dilakukan secara "stream", atau terus-menerus. Dari sudut pandang prosesnya, *streaming* berarti sebuah teknologi pengiriman *file* dari *server* ke *client* melalui jaringan *packet-based* misalnya internet. Sedangkan dari sudut pandang *user*, *streaming* adalah teknologi yang memungkinkan suatu *file* dapat segera dijalankan tanpa harus menunggu selesai diunduh seluruhnya dan terus mengalir tanpa ada interupsi.

File yang dapat ditransmisikan oleh video *streaming* adalah *file* audio, video, *image*, *text*, data 3D, *software*, dan sebagainya. Tetapi *streaming* lebih mengacu kepada *time based* media, khususnya audio dan video, yang harus dapat dinikmati sesegera mungkin dan berdasarkan pada pewaktuan yang tepat, karena untuk dapat menikmati lagu atau film, haruslah dimainkan secara berurutan dari awal hingga akhir (*sequential*) tanpa terputus-putus (*uninterrupted*). Ada beberapa jenis video *streaming* seperti ditunjukkan pada Tabel 2.2 berikut:





Tabel 2.2 : Jenis-jenis Video Streaming

No.	Jenis Video	Penjelasan
1.	<i>Live streaming</i>	Terjadi ketika setiap <i>user</i> melihat video <i>streaming</i> yang sama secara bersamaan
2.	<i>On-demand streaming</i>	Terjadi saat pengguna dapat meminta <i>prerecorded</i> video dialirkan kepada pengguna ketika pengguna ingin melihat video tersebut. Dalam hal ini, setiap <i>user</i> dapat memilih kapan saat untuk memulai dan mengakhiri <i>streaming</i> video.
3.	<i>Download and play</i>	Menggunakan memori atau hard disk di dalam perangkat <i>user</i> untuk menerima file audio/video yang kemudian dapat ditampilkan (dimainkan) oleh perangkat <i>user</i> . Salah satu keuntungan utama dari <i>download and play</i> adalah bahwa file audio/video dapat dikirimkan dengan HTTP, sehingga kemungkinan dapat melewati firewall lebih besar. Beberapa <i>server</i> tidak setuju dengan <i>download and play</i> , karena hasil salinan video tersimpan di perangkat penyimpanan setiap <i>user</i> sebelum video dilihat/ditampilkan.
4.	<i>Progressive download and play</i>	<i>Progressive download and play</i> adalah salah satu metode yang paling populer untuk pengiriman file video/audio melalui Internet, dan digunakan oleh situs-situs seperti <i>YouTube</i> . Teknologi ini pada dasarnya sama seperti <i>download and play</i> , kecuali file video/audio dipecah menjadi <i>file-file</i> kecil yang dikirimkan sebagai video <i>playback</i> . <i>Progressive download and play</i> memiliki keuntungan, yaitu dapat melewati <i>firewall</i> dengan mudah ketika mentransmisikan kecepatan dan kebutuhan <i>storage</i> yang lebih kecil.

(Sumber: Wes Simpson, 2008: 41)

Ada beberapa bentuk komunikasi pada video *streaming*, yaitu (Simpson, 2008) :

- a. *Broadcast* adalah bentuk komunikasi *one to many* (yang pada dasarnya adalah *one to all*) yang paling dikenal, salah satu contohnya adalah siaran TV. Dengan menggunakan *broadcast* maka informasi yang akan dikirimkan berasal dari satu sumber atau titik kepada semua penerima yang yang tergabung dalam jaringan. Pada *broadcast* semua penerima mau tidak mau akan menerima informasi ini.
- b. *Multicast*, merupakan bentuk komunikasi *one to many*, tetapi tidak seperti *broadcast*. Pada *multicast*, informasi yang dikirimkan berasal dari satu sumber atau titik kepada semua penerima yang menginginkan informasi tersebut. *Server* akan membuat *stream* satu kali kemudian *stream* ini diduplikasi dan dikirimkan ke setiap *client*. Setiap *client* akan menerima *stream* yang sama dengan *client* lainnya. Salah satu contoh penggunaan *multicast* adalah *live video*.
- c. *Unicast* merupakan bentuk komunikasi *one to one* atau *point to point*. Pada *unicast*, informasi yang dikirimkan berasal dari satu sumber atau titik ke satu titik lainnya. *Server* akan mengirimkan file *streaming* ke komputer *client* berkali-kali bergantung pada banyaknya jumlah permintaan. Setiap *client* akan menerima file *streaming* yang terpisah dari *client* lainnya. Contoh penggunaan *unicast* adalah *video on demand*.

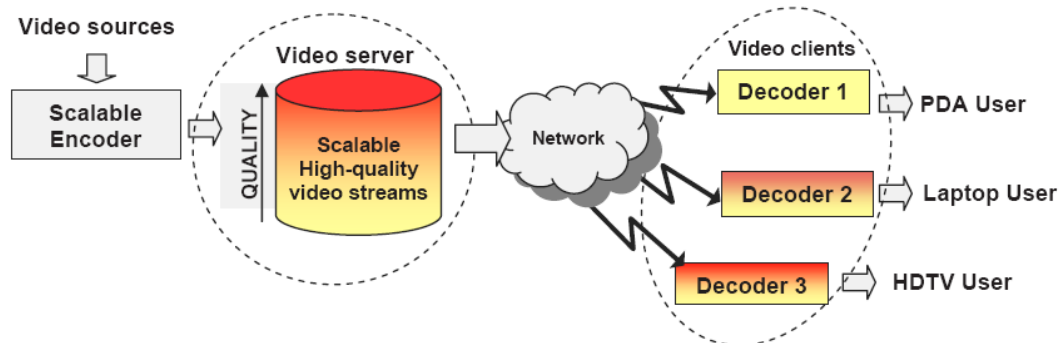
### 2.3.1 CODEC (Coder/DECoder)

Pada aplikasi multimedia baik untuk *voice*, video, maupun data, besarnya *bandwidth* tergantung dari CODEC yang digunakan. CODEC terdiri *Encoder* yang digunakan untuk melakukan *encoding* pada file audio atau video sehingga memiliki ukuran yang lebih kecil dengan memperkecil *pixel* dan *frame* serta *Decoder* yang digunakan untuk membaca file yang telah di-*encode* dan memainkannya di sisi *client*. CODEC sangat diperlukan dalam proses video *streaming*.

Pada proses *streaming* video, video dari *source* akan di-*capture* dan disimpan pada sebuah *buffer* yang berada pada memori komputer (bukan media penyimpanan seperti *harddisk*) dan kemudian di-*encode* sesuai dengan format yang diinginkan. Dalam proses *encode* ini, *server* dapat mengompresi data sehingga ukurannya tidak terlalu besar (bersifat *optional*). Hal tersebut dilakukan karena keterbatasan *bandwidth* jaringan. Setelah di-*encode*, data akan di-*stream* ke *user* yang lain. *User* akan



melakukan *decode* data dan menampilkan hasilnya ke layar *user* untuk menikmati video *streaming*. Proses video *streaming* dapat ditunjukkan pada Gambar 2.7.



**Gambar 2.7** : Proses *streaming* video melalui jaringan  
(Sumber: Ashraf M.A. Ahmad and Ismail Khalil Ibrahim, 2009: 94)

Terdapat dua jenis CODEC yang dapat digunakan dalam video *streaming*, yaitu (Andreas Handojo, dkk, 2009: 1):

- a. *Lossy* CODEC: CODEC ini akan mengurangi kualitas data dengan mengurangi ukuran data (kompresi). Pada umumnya, CODEC ini digunakan untuk menyimpan data pada media penyimpanan yang berukuran terbatas seperti CD-ROM dan DVD. *Lossy* CODEC ini biasanya digunakan untuk *streaming*, karena *bandwidth* jaringan yang terbatas. Contoh: Windows Media Video, H.264.
- b. *Lossless* CODEC: Pada *lossless* CODEC ini, kualitas data yang dihasilkan tidak akan berkurang. Tetapi, ukuran data yang dihasilkan oleh *lossless* CODEC ini akan lebih besar dibandingkan dengan *lossy* codec. Pada umumnya, *lossless* CODEC ini digunakan pada video yang masih memerlukan *editing*, karena dalam proses *editing* dilakukan *encode-decode* berulang kali, sehingga jika menggunakan *lossy* CODEC, kualitas video akan jauh menurun dibandingkan dengan video aslinya. Contoh: CorePNG, huffyuv, Apple Lossless Audio Codec.

Ada beberapa jenis audio dan video CODEC menurut ITU.T, seperti ditunjukkan pada Tabel 2.3.

Tabel 2.3 : Jenis CODEC Video dan Audio

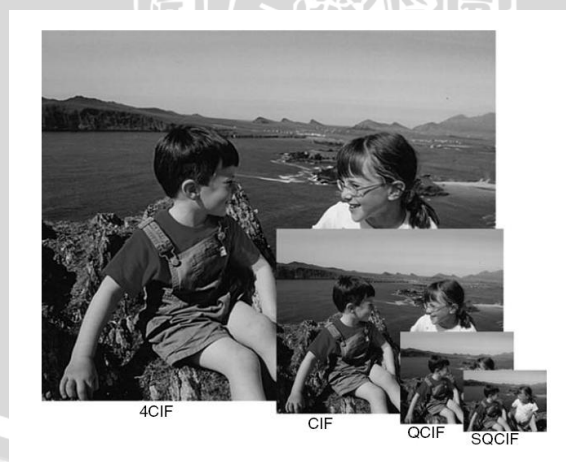
Audio CODEC	Bit Rate (kbps)	Maximum Payload (byte)	Delay CODEC (ms)
AMR-WB	6,6 – 23,85	35	10 – 20
AMR-WB+	5,2 – 48	46	20 – 40
HE-AAC v2	128 – 320	80	40 – 80
Video CODEC	Bit Rate (kbps)	Maximum Payload (byte)	Delay CODEC (ms)
H.264/AVC	64 – 384	254	150 – 300

(Sumber: RFC 4352 and RFC 3984 RTP Payload Format for H.264 Video, 2005)

### 2.3.2 Kualitas Video

Video dapat juga disebut sebagai Gambar-Gambar yang bergerak. Dalam video tampilkan sejumlah Gambar atau *frame* dengan kecepatan tertentu yang disebut dengan istilah *frame rate*, yang dihitung dalam skala *frame per second* (fps). Seperti jenis data yang lain, data video juga dapat disimpan, diedit, ataupun dikirim melalui jaringan.

Ada beberapa format Gambar yang digunakan dalam aplikasi video *streaming* diantaranya adalah format CIF, QCIF, SQCIF dan 4CIF. Perbandingan untuk setiap format Gambar dapat dilihat pada Gambar 2.8.



Gambar 2.8 : Format Gambar  
(Sumber: Richardson, 2003: 20)

Pilihan resolusi *frame* tergantung pada aplikasi dan kapasitas yang tersedia atau kapasitas transmisi. Sebagai contoh, 4CIF sesuai untuk digunakan pada televisi standar dan DVD-video. Format CIF dan QCIF sangat populer digunakan pada aplikasi video



*conferencing*. Sedangkan format QCIF atau SQCIF sesuai untuk digunakan pada aplikasi *mobile multimedia* dimana resolusi layar dan *bit rate* format ini terbatas.

Kualitas video yang baik dapat dinilai dari tiga elemen utama, yaitu: (Richardson, 2003):

- a. *Frame Rate*, jumlah Gambar yang ditampilkan per detik pada video. Minimum frame yang dibutuhkan untuk mendapatkan ilusi gambar yang bergerak adalah sekitar 15 fps.
- b. *Color Depth*, Jumlah *bit* pada setiap *pixel* yang menunjukkan informasi warna. Misalnya: 24 *bit* menunjukkan 16,7 juta warna, 16 *bit* sekitar 65.535 warna, atau 8*bit* hanya 256 warna.
- c. *Frame Resolution*, biasanya ditunjukkan dengan *width* dan *height* pada *pixel*. Misalnya: *full screen* PC mempunyai *frame resolution* sebesar 640x480. Jika *user* menggunakan kecepatan pengiriman dalam *frame per second* (fps) video yang rendah, akan memakan bandwidth yang lebih rendah dibandingkan *frame per second* (fps) yang tinggi. Video yang cukup baik biasanya dikirim dengan kecepatan *frame per second* (fps) sekitar 30 fps.

### 2.3.3 Enkapsulasi Data IP

Pada IP terjadi penyampaian data dari protokol di satu *layer* ke protokol di *layer* lain. Setiap protokol pada masing-masing *layer* akan menambahkan informasi tambahan miliknya pada data jika protokol tersebut menerima data dari protokol lain di atasnya. Tambahan informasi tersebut dinamakan dengan *header*, yang berfungsi sebagai kontrak informasi. Proses penambahan informasi ini disebut dengan *enkapsulasi*.

#### a) RTP (*Real Time Protocol*)

RTP adalah *IP-based protocol* yang menyediakan dukungan *transport data* antara *real-time* seperti video dan audio *stream*, termasuk pada layanan video *streaming*. RTP digunakan untuk transport protokol dengan menggunakan UDP, TCP, ST-2, atau yang lainnya. RTP secara tipikal berada di atas IP untuk penggunaan dari *multiplexing* dengan fungsi *checksum*. UDP dipilih sebagai *protocol transport* untuk RTP karena RTP didesain untuk *multicast*, *connection oriented* TCP tidak memiliki skala yang baik. Selain itu untuk *real-time data*, *reliability* tidak penting untuk pengiriman secara waktu.

b) UDP (*User Datagram Protocol*)

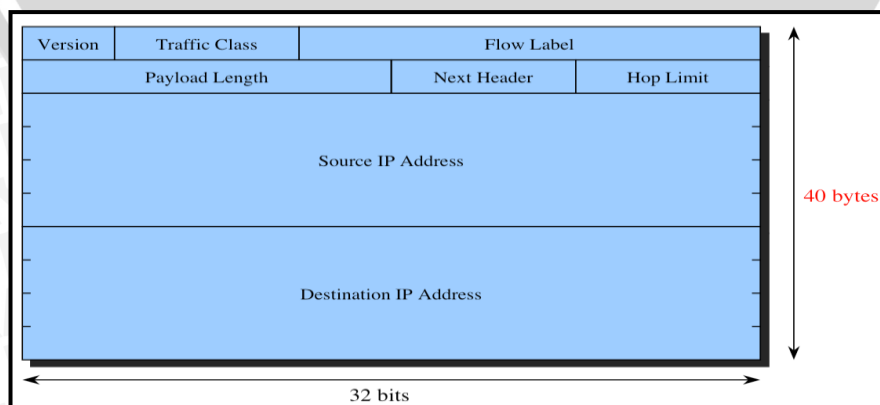
UDP merupakan protokol yang berada di lapisan transport IP *layer*. Protokol ini bersifat *connectionless* dan *unreliable*. *Connectionless* berarti tidak diperlukan pembentukan hubungan terlebih dahulu untuk mengirimkan data. *Unreliable* berarti data tidak dijamin akan sampai di tujuan yang benar dan dalam kondisi yang benar pula. Protokol *transport* UDP lebih sering diimplementasikan untuk aplikasi yang mengarah pada proses *real-time*, seperti aplikasi multimedia pada layanan video *streaming*, dimana rugi-rugi paket data yang lebih kecil akan lebih ditoleransi daripada nilai *delay* yang terjadi pada IP.

c) IP (*Internet Protocol*)

Protokol IP merupakan inti dari protokol TCP/IP, yang bersifat *unreliable*, *connectionless*, dan *datagram delivery service*. Banyak yang mengistilahkan sebagai *best effort delivery*, yang artinya IP menyediakan *no error checking* atau *tracking*. Jika diperlukan reliabilitas, maka IP harus dipasangkan dengan protokol yang *reliable*, misalnya TCP. IP mempunyai fungsi utama yaitu *connectionless-service*, pemecahan (*fragmentasi*) dan penyatuan dari paket-paket, serta *routing*.

• **Format Header Ipv6**

Pada Jaringan UMTS dan LTE menggunakan IP versi 6. *Header* IPv6 didesain mempunyai panjang *header* 40 *byte* dan memiliki *field* lebih sedikit dibandingkan dengan IPv4, sehingga *router* melakukan penanganan tiap *field* lebih sedikit menyebabkan proses *routing* atau aliran data menjadi lebih ringkas. Format *header* alamat IPv6 ditunjukkan dalam Gambar 2.9.



**Gambar 2.9** : Format *Header* Ipv6  
(Sumber : Lain E.G, Richardson. 2003)



d) H.264/AVC

H.264/AVC merupakan standar pengkodean video terbaru dari ITU-T *Video Coding Expert Group* dan ISO/IEC *Moving Picture Expert Group*. Tujuan utama dari standarisasi H.264/AVC untuk meningkatkan performa kompresi dan penyediaan sebuah penyajian video yang “*network-friendly*” pada aplikasi video telepon, *Mobile TV*, penyimpanan, penyiaran atau *streaming*.

e) AMR-WB +(Adaptive Multi Rate – WideBand +)

*Adaptive Multi Rate – WideBand* (AMR-WB+) adalah standar pengkodean audio setelah AMR yang menggunakan teknologi yang sama seperti *Algebraic Code Linear Excited Prediction* (ACLEP).

## 2.4 Handover

*Handover* adalah proses pengalihan antara satu kanal terhadap kanal yang lain. *Handover* diperlukan sistem seluler untuk menjamin adanya kontinuitas komunikasi apabila pelanggan bergerak dari satu *cell* ke *cell* lain. Proses ini sangat penting dalam komunikasi seluler melihat latar belakang pemakaian telepon seluler yang penuh mobilitas, jumlah trafik, dan kualitas sinyal, sehingga tidak terjadi *drop call* atau terputusnya hubungan.

*Handover* dapat terjadi dalam kaitannya dengan pergantian sumber radio yang menyediakan suatu layanan dengan melakukan perubahan yang melibatkan *base station*. Hal ini terjadi khususnya ketika sumber radio menyediakan pergantian layanan dari *mode* akses radio UTRA (*UMTS Terrestrial Radio Acces*) yang satu ke UTRA yang lainnya.

### 2.4.1 Jenis Handover pada Sistem Komunikasi Bergerak

Dengan semakin berkembangnya teknologi *wireless selular*, maka definisi mengenai proses *handover* menjadi semakin meluas tidak sebatas untuk satu jenis teknologi yang digunakan. Oleh sebab itu *handover* sendiri dapat dikategorikan dalam beberapa jenis yang dapat dijelaskan sebagai berikut : (Areej Saleh, 2004 : 17)

#### 2.4.1.1 Berdasarkan Teknologi yang Digunakan:

- a. *Horizontal Handover*, adalah proses *handover* yang terjadi pada sebuah *User Equipment* (UE) yang mengalami perpindahan antar *base station* dengan teknologi yang sama, hal ini juga sering dikenal sebagai *intra-technology handoff*.

- b. *Vertical Handover*, adalah proses *handover* yang terjadi pada sebuah UE yang melibatkan dua *interface* jaringan yang berbeda dengan teknologi yang berbeda pula. *Handover* ini juga dikenal sebagai *inter-technology handoff*.

#### 2.4.1.2 Berdasarkan Domain Administratif:

- a. *Intra-Administrative Domain Handover*, adalah proses *handover* yang terjadi pada sebuah UE yang mengalami perpindahan antar *base station* dengan teknologi yang sama maupun berbeda namun keduanya dikelola dalam satu *domain* administratif.
- b. *Inter-Administrative Domain Handover*, adalah proses *handover* yang terjadi pada sebuah UE yang mengalami perpindahan antar *base station* dengan teknologi yang sama ataupun berbeda namun keduanya dikelola dalam *domain* administratif yang berbeda.

#### 2.4.1.3 Berdasarkan Konektivitas:

- a. *Hard Handover*, terjadi pada sebuah UE dengan melakukan pemutusan hubungan dengan *base station* yang sebelumnya untuk kemudian melakukan hubungan baru dengan *base station* pada sel yang lain, proses ini lebih dikenal dengan "*break before make*".
- b. *Soft Handover*, terjadi pada sebuah UE yang melakukan hubungan baru terlebih dahulu saat melakukan perpindahan sel dengan maksud untuk menjaga konektivitasnya, atau dikenal dengan "*make before break*".
- c. *Softer Handover*, *handover* ini hampir mirip dengan *soft handover*, namun UE mengalami perpindahan konektivitas *radio link* dalam sebuah *base station* yang sama.

#### 2.4.1.4 Berdasarkan Frekuensi yang digunakan:

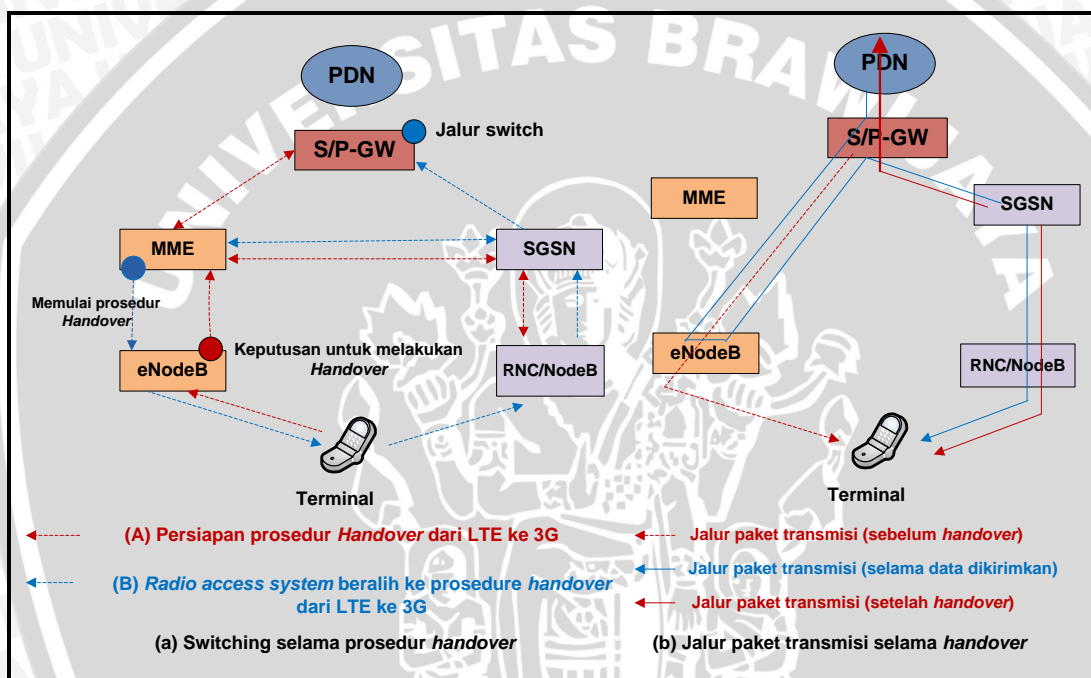
- a. *Intra-Frequency Handover*, adalah proses *handover* yang terjadi pada UE yang mengalami perpindahan antar sel dengan frekuensi kerja yang sama.
- b. *Inter-Frequency Handover*, adalah proses *handover* yang terjadi pada UE yang mengalami perpindahan antar sel dengan frekuensi kerja yang berbeda.

#### 2.4.2 Handover Jaringan LTE ke UMTS

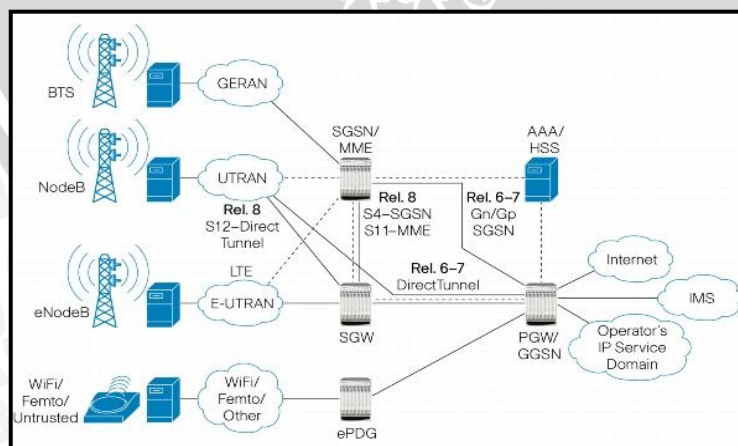
*Handover* ini diperlukan oleh pelanggan LTE yang bergerak dari satu sel dengan jaringan LTE ke suatu sel tanpa jaringan LTE. *Handover* ini sering juga disebut sebagai *Intersystem Handover LTE to UMTS*. Dengan EPC (*Evolved Packet Core*),



memungkinkan komunikasi akan berjalan terus menerus, bahkan ketika terminal beralih dari satu jenis sistem radio dengan sistem radio yang lainnya. Khususnya, dalam rangka mencapai jalur jaringan internal *switching* maka diperlukan mengubah akses sistem radio, S-GW menyediakan manajemen mobilitas yang berfungsi untuk *handover* antara sistem akses radio 3GPP, dan P-GW berfungsi menyediakan layanan *handover* antara 3GPP dan non-sistem akses radio 3GPP. Dengan cara ini, alamat IP tidak berubah saat terminal switch mengakses sistem radio, sehingga komunikasi dapat berjalan terus setelah proses *handover*.



**Gambar 2.10** : Proses *Handover* dari LTE ke 3G  
 (Sumber : Nishida, 2009, NTT DOCOMO Technical journal Vol.11 No.3)

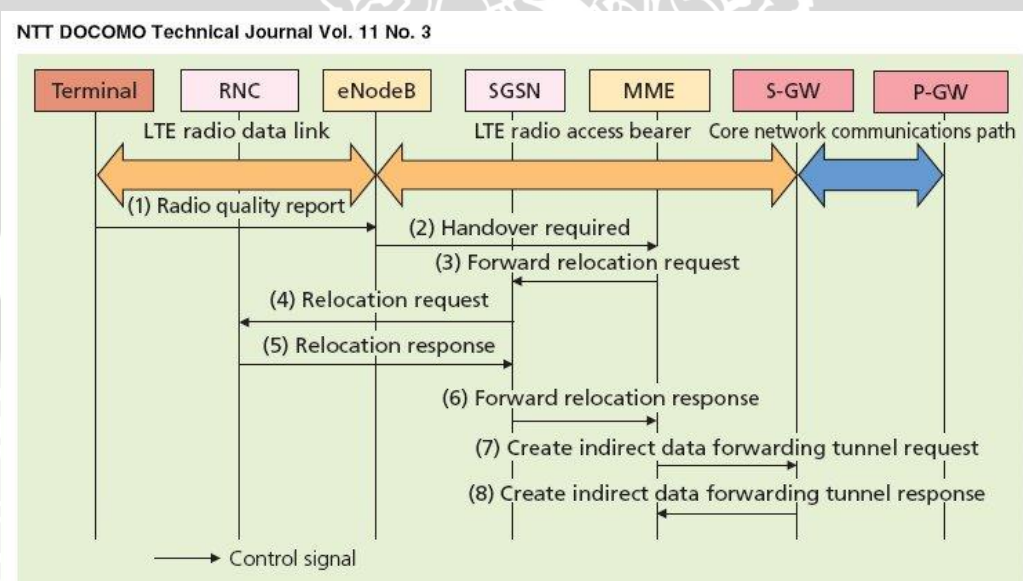


**Gambar 2.11** : Intersystem *Handover* LTE to UMTS  
 (Sumber : CISCO systems, white paper 2010)

Dalam proses *handover* antara sistem akses radio 3GPP LTE dan 3G, persiapan *handover* dilakukan sebelum mengubah sistem, termasuk tugas-tugas seperti mengamankan sumber daya pada sistem radio, melalui kerjasama antara sistem akses radio (Gambar 2.10(a) (A)). Kemudian, ketika *switch* aktif, hanya mengaktifkan jalur jaringan yang diperlukan, sehingga mengurangi waktu proses *handover* (Gambar 2.10 (a) (B)). Selain itu, hilangnya paket data yang datang di jalur akses *pra-switch* selama *handover* dapat dihindari dengan menggunakan fungsi data *forwarding* (Gambar 2.10 (b)). Dengan cara ini, melalui interaksi antara sistem radio, memungkinkan *handover* semakin cepat, bahkan antara sistem akses radio seperti LTE dan 3G yang tidak dapat digunakan secara bersamaan.

#### 2.4.2.1 Prosedur Persiapan *handover*

Prosedur persiapan *handover* untuk beralih akses radio dari LTE ke 3G yang ditunjukkan pada (Gambar 2.10 (a) (A)) dapat dikembangkan kembali pada Gambar 2.12 berikut.



**Gambar 2.12** : Prosedur Persiapan *handover* dari LTE ke 3G  
(Sumber : Nishida, 2009, NTT DOCOMO Technical journal Vol.11 No.3)

Langkah-langkah Prosedur Persiapan *Handover* dari LTE ke 3G :

Langkah (1) : Terminal mengirimkan laporan kualitas radio yang merupakan awal dari *handover*. eNodeB memutuskan apakah *handover* harus dilakukan berdasarkan informasi dalam laporan, juga mengidentifikasi *base station* dan RNC untuk beralih, dan mulai persiapan *handover*.



Langkah (2) (3): eNodeB yang mengirimkan kebutuhan *handover* ke MME, kemudian mengirimkan RNC *identifier* dan informasi kontrol transmisi untuk sistem radio yang akan dituju. MME mengidentifikasi apakah SGSN terhubung ke RNC target berdasarkan diterimanya RNC *identifier* dan mengirim kontrol komunikasi dan informasi lain yang diterima dari eNodeB ke SGSN dalam sinyal relokasi permintaan berikutnya. Informasi yang diperlukan untuk mengkonfigurasi jalur komunikasi antara S-GW dan SGSN, yang mana digunakan untuk transmisi data setelah MME menyelesaikan *handover*, dikirim pada waktu yang bersamaan.

Langkah (4) (5): SGSN akan meneruskan permintaan relokasi ke RNC, memberitahu informasi tersebut dari kontrol informasi komunikasi yang ditularkan dari eNodeB. RNC melakukan konfigurasi pengolahan radio yang diperlukan berdasarkan informasi yang diterima dan mengirimkan respon relokasi ke SGSN. Perhatikan bahwa melalui proses ini, radio pembawa pada 3G disusun antara SGSN dan RNC.

Langkah (6): SGSN mengirimkan *forward* respon relokasi ke MME untuk memberitahu MME bahwa prosedur relokasi telah selesai. Sinyal ini juga mencakup data yang dikeluarkan oleh SSGN dan diperlukan untuk mengkonfigurasi jalur komunikasi dari S-GW ke SGSN yang akan digunakan untuk *forwarding* data.

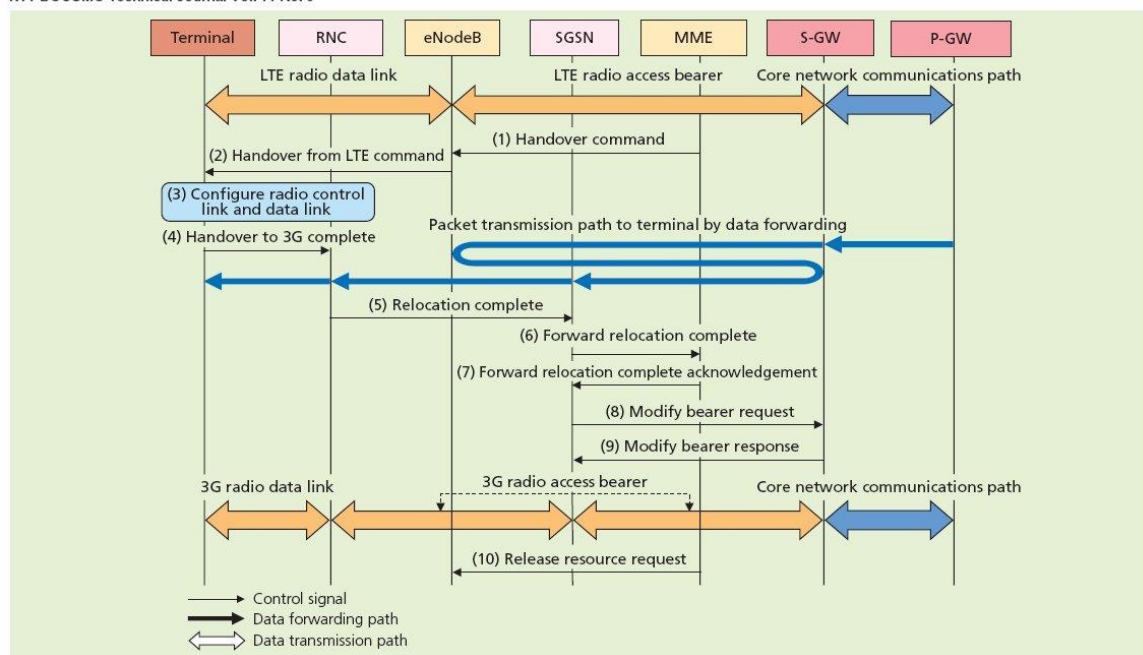
Langkah (7) (8): MME mengirimkan permintaan pembuatan jalur ke S-GW, dan informasi yang berasal dari SSGN itu hanya diterima. Informasi yang diterima S-GW membentuk jalur komunikasi menuju SGSN untuk meneruskan pengiriman data dan membuat data mudah diteruskan ke MME.

Melalui persiapan *handover*, radio 3G yang menjadi tujuan akan mengakses sumber daya yang akan diperlukan, kemudian akses radio antara SGSN dan RNC dikonfigurasi, dan meneruskan jalur data dari S-GW ke SGSN.

#### **2.4.2.2 Prosedur Handover untuk Switching Radio Acces System**

Setelah prosedur persiapan *handover* untuk beralih akses radio dari LTE ke 3G yang ditunjukkan pada Gambar 2.12 selesai dilakukan, maka langkah-langkah *switching* radio pada saat *handover* dari jaringan LTE ke 3G dapat dilakukan seperti Gambar 2.13 berikut.

NTT DOCOMO Technical Journal Vol. 11 No. 3



**Gambar 2.13** : Prosedur *Switching* sistem radio pada *handover* jaringan LTE ke 3G  
(Sumber : Nishida, 2009, NTT DOCOMO Technical journal Vol.11 No.3)

Langkah-langkah *switching* radio pada saat *handover* dari jaringan LTE ke 3G :

Langkah (1) ke (2) : Saat persiapan *handover* yang digambarkan pada Gambar 2.12 selesai, MME mengirim sebuah perintah *handover* ke eNodeB. Ketika menerima sinyal, eNodeB mengirim sebuah perintah *handover* dari LTE untuk mengalihkan sistem radio. Perlu dicatat bahwa pada saat eNodeB menerima perintah *handover* dari MME, ia mulai meneruskan paket data yang diterima dari S-GW. Setelah itu, paket untuk terminal yang sampai pada S-GW diteruskan ke terminal berdasarkan jalur: S-GW, eNodeB, S-GW, SGSN, RNC.

Langkah (3) ke (6) : Terminal beralih ke 3G dan ketika konfigurasi radio *link* selesai, pemberitahuan bahwa telah tersambung ke sistem radio *aces* 3G dikirim menuju masing - masing *link* melalui MME yaitu dari terminal ke RNC, dari RNC ke SGSN, dan dari SGSN ke MME. Pada cara ini, MME dapat melakukan langkah (10), yang akan dijelaskan di bawah, untuk melepaskan sumber eNodeB setelah satu set periode waktu terlewati.

Langkah (7) : MME mengirim balasan sinyal ACK ke SGSN. Satu set periode waktu setelah meneria sinyal, SGSN melepaskan sumber yang terkait dengan pengiriman data.

Langkah (8) : SGSN mengirimkan permintaan sinyal baru ke S-GW untuk mengubah dari jalur komunikasi sebelum *handover*, antara S-GW dan eNodeB, dan



diantara S-GW dan SGSN. Sinyal ini mengandung unsur-unsur informasi yang dibutuhkan untuk mengkonfigurasi bagian dari S-GW ke SGSN, termasuk yang dikeluarkan oleh SGSN. Ketika S-GW menerima sinyal ini, mengkonfigurasi jalur komunikasi dari S-GW ke SGSN. Dengan cara ini, jalur komunikasi menjadi: S-GW, SGSN, RNC, terminal, dan transmisi data dengan sistem akses radio 3G target dimulai.

Catatan : setelah poin ini data tidak dibutuhkan lagi, jadi S-GW mengirim paket ke eNodeB dengan sebuah “*end marker*” yang ada, dan ketika eNodeB menerima paket ini, akan melepas sumbernya yang berkaitan dengan data *forwarding*.

Langkah (9) sampai (10) : S-GW mengirim respon pembawa yang telah dimodifikasi ke SGSN, dan nantinya akan menunjukkan bahwa prosedur *handover* telah selesai. MME juga melepaskan eNodeB yang tidak digunakan lagi.

Melalui prosedur *handover* ini data diteruskan saat proses *handover*, penyambungan akses dari radio telah selesai dan komunikasi dari P-GW ke terminal telah di perbarui.

Pada Gambar 2.13, diterangkan bahwa prosedur *handover* 3GPP sistem radio pada S-GW tidak berubah, tetapi proses *handover* dengan relokasi S-GW masih dimungkinkan. Pada kasus ini, P-GW menyediakan jalannya *switching*, seperti halnya pada akses yang beralih ke sistem akses non-3GPP.

## 2.5 Parameter Performansi *Handover* dari Jaringan LTE ke UMTS

### 2.5.1 Paket Data Aplikasi *Video Streaming*

Pada aplikasi *video streaming*, paket yang ditransmisikan dibedakan atas paket audio dan paket video, dimana tiap paket tersebut mempunyai besar *payload* yang berbeda. Aplikasi *video streaming* menggunakan jenis CODEC H.264/AVC untuk video dengan *bit rate* CODEC antara 64 – 384 kbps dan AMR-WB+ untuk audio dengan *bit rate* CODEC antara 5,2 – 48 kbps. Besar *payload* tiap paket audio dan video dapat dinyatakan dengan persamaan:

$$PL_a = B_{\text{CODEC}_a} \times f_R \quad (2-1)$$

$$PL_v = B_{\text{CODEC}_v} \times f_R \quad (2-2)$$

dengan:

$PL_a$  = *payload* paket audio (bit)

$PL_v$  = *payload* paket video (bit)

$B_{\text{CODECA}} = \text{bit rate CODEC audio (bps)}$

$B_{\text{CODECV}} = \text{bit rate CODEC video (bps)}$

$f_{\text{R}} = \text{frame rate (s)}$

Besarnya tiap paket audio dan video aplikasi video *streaming* merupakan penjumlahan dari *payload* paket audio dan video dengan *header* RTP, UDP, dan IP. Sehingga, besar tiap paket audio dan video *streaming* yang dihasilkan dapat dinyatakan dengan (Wes Simpson, 2008):

$$P_a = \text{PLa}/\text{PLa}_{\text{max}} \quad (2-3)$$

$$P_a\text{-size} = \text{PLa} + P_a \times (\text{H}_{\text{RTP}} + \text{H}_{\text{UDP}} + \text{H}_{\text{IP}}) \quad (2-4)$$

$$P_v = \text{PLv}/\text{PLv}_{\text{max}} \quad (2-5)$$

$$P_v\text{-size} = \text{PLv} + P_v \times (\text{H}_{\text{RTP}} + \text{H}_{\text{UDP}} + \text{H}_{\text{IP}}) \quad (2-6)$$

dengan:

$P_a = \text{jumlah paket audio}$

$\text{PLa}_{\text{max}} = \text{maksimum payload paket audio (bit)}$

$P_v = \text{jumlah paket video}$

$\text{PLv}_{\text{max}} = \text{maksimum payload paket video (bit)}$ dimana:

Paket-paket video dan audio yang telah di-*encode* akan ditransmisikan melalui RTP (*Real Time Protocol*). Besar paket video *streaming* yang akan ditransmisikan dapat diperoleh menggunakan persamaan (Wes Simpson, 2008):

$$P_{\text{vs-size}} = P_a\text{-size} + P_v\text{-size} \quad (2-7)$$

dengan:

$P_{\text{vs-size}} = \text{ukuran paket aplikasi video streaming (byte)}$

$P_a\text{-size} = \text{besar paket audio (byte)}$

$P_v\text{-size} = \text{besar paket video (byte)}$

$\text{H}_{\text{RTP}} = \text{header RTP (12 byte)}$

$\text{H}_{\text{IP}} = \text{panjang header IP (40 byte)}$

$\text{H}_{\text{UDP}} = \text{panjang header UDP (8 byte)}$



### 2.5.2 Delay

Definisi umum dari *delay* adalah waktu tunda. Pada Aplikasi video *streaming* *delay-delay* tersebut sangat berpengaruh pada kualitas layanan. Semakin kecil *delay* yang dihasilkan akan semakin bagus pula kualitas layanan video *streaming* tersebut:

#### 2.5.2.1 Delay end-to-end

Dalam jaringan yang berbasiskan *packet switching*, *delay* yang terjadi merupakan penjumlahan *delay-delay* yang ada dalam perjalanan paket dari sumber ke tujuannya. *Delay end-to-end* dapat dihitung dengan menggunakan persamaan berikut (Anurag Kumar, 2008 : 65) :

$$t_{net} = t_{proc} + t_T + t_P + t_W \quad (s) \quad (2-8)$$

Dengan :

$t_{net}$  = *delay end-to-end* pada jaringan LTE dan UMTS (s)

$t_{proc}$  = *delay* proses (s)

$t_T$  = *delay* transmisi (s)

$t_P$  = *delay* propagsi (s)

$t_W$  = *delay* antrian (s)

Aplikasi video *streaming* merupakan salah satu teknologi telekomunikasi yang sangat sensitif terhadap *delay*. Sesuai dengan referensi International Telecommunications Union (ITU) G.110, batasan *delay* untuk aplikasi video *streaming* adalah < 10s seperti yang ditunjukkan pada Gambar 2.14.

Error tolerant	Conversational voice and video	Voice/video messaging	Streaming audio and video	Fax
Error intolerant	Command/control (e.g. Telnet, interactive games)	Transactions (e.g. E-commerce, WWW browsing, Email access)	Messaging, Downloads (e.g. FTP, still image)	Background (e.g. Usenet)
	Interactive (delay <<1 s)	Responsive (delay ~2 s)	Timely (delay ~10 s)	Non-critical (delay >>10 s)

T1213060-02

**Gambar 2.14** Batasan *Delay* setiap Aplikasi Multimedia (Sumber: ITU.T G.1010, 2001: 14)

### 2.5.2.2 Delay Proses

*Delay* proses merupakan waktu yang dibutuhkan untuk memproses paket data dan untuk menentukan kemana data tersebut akan diteruskan. *Delay* proses pada jaringan UMTS dan LTE meliputi *delay* enkapsulasi dan *delay* dekapsulasi.

*Delay* enkapsulasi adalah waktu yang dibutuhkan untuk menambahkan keseluruhan *header* pada sebuah paket sehingga paket data tersebut dapat tepat sampai ke tujuan. Sedangkan *delay* dekapsulasi adalah waktu yang dibutuhkan untuk melepaskan keseluruhan *header* dari sebuah paket. Besarnya *delay* dekapsulasi dan *delay* enkapsulasi dapat diperoleh dengan menggunakan persamaan di bawah ini (Uke Kurniawan, 2010:40):

$$t_{D(mode)} = \frac{W_{node} \times 8}{C_{node}} \quad (2-9)$$

$$t_{E(mode)} = \frac{W_{node} \times 8}{C_{(node)}} \quad (2-10)$$

dengan:

$t_e$  = *delay* enkapsulasi (s)

$t_d$  = *delay* dekapsulasi (s)

$W_{node}$  = panjang *frame* (byte)

$C$  = kecepatan pemrosesan data (bps)

*Delay* enkapsulasi terjadi di setiap *node* pada jaringan UMTS yaitu *server*, GGSN, SGSN, dan RAN. Besarnya *delay* enkapsulasi didapatkan dengan menggunakan persamaan:

$$t_{enc} = t_{E1} + t_{E2} + t_{E3} + t_{E4} \quad (2-11)$$

dengan:

$t_{enc}$  = *delay* enkapsulasi total

$t_{E1}$  = *delay* enkapsulasi pada *server*

$t_{E2}$  = *delay* enkapsulasi pada GGSN

$t_{E3}$  = *delay* enkapsulasi pada SGSN

$t_{E4}$  = *delay* enkapsulasi pada RAN

Sedangkan *delay* dekapsulasi terjadi di setiap *node* pada jaringan UMTS, yaitu GGSN, SGSN, RAN, dan UE. Besarnya *delay* dekapsulasi dapat dihitung menggunakan persamaan:



$$t_{dec} = t_{D1} + t_{D2} + t_{D3} + t_{D4} \quad (2-12)$$

dengan:

$t_{dec}$  = *delay* dekapsulasi total

$t_{D1}$  = *delay* dekapsulasi pada GGSN

$t_{D2}$  = *delay* dekapsulasi pada SGSN

$t_{D3}$  = *delay* dekapsulasi pada RAN

$t_{D4}$  = *delay* dekapsulasi pada UE

Nilai *delay* proses pada jaringan UMTS untuk aplikasi video *streaming* dapat diperoleh dengan menggunakan persamaan:

$$t_{proc} = t_{enc} + t_{dec} \quad (2-13)$$

dengan:

$t_{proc}$  = *delay* proses

$t_{enc}$  = *delay* enkapsulasi total

$t_{dec}$  = *delay* dekapsulasi total

*Delay* enkapsulasi terjadi di setiap *node* pada jaringan LTE yaitu *server*, PDN-GW, S-GW, dan eNodeB. Besarnya *delay* enkapsulasi didapatkan dengan menggunakan persamaan:

$$t_{enc} = t_{E1} + t_{E2} + t_{E3} + t_{E4} \quad (2-14)$$

dengan:

$t_{enc}$  = *delay* enkapsulasi total

$t_{E1}$  = *delay* enkapsulasi pada *server*

$t_{E2}$  = *delay* enkapsulasi pada PDN-GW

$t_{E3}$  = *delay* enkapsulasi pada S-GW

$t_{E4}$  = *delay* enkapsulasi pada eNodeB

Sedangkan *delay* dekapsulasi terjadi di setiap *node* pada jaringan LTE, yaitu PDN-GW, S-GW, eNodeB dan UE. Besarnya *delay* dekapsulasi dapat dihitung menggunakan persamaan:

$$t_{dec} = t_{D1} + t_{D2} + t_{D3} + t_{D4} \quad (2-15)$$

dengan:

$t_{dec}$  = *delay* dekapsulasi total

$t_{D1}$  = *delay* dekapsulasi pada PDN-GW

$t_{D2}$  = *delay* dekapsulasi pada S-GW

$t_{D3}$  = *delay* dekapsulasi pada *eNodeB*

$t_{D4}$  = *delay* dekapsulasi pada UE

Nilai *delay* proses pada jaringan LTE untuk aplikasi video *streaming* dapat diperoleh dengan menggunakan persamaan:

$$t_{proc} = t_{enc} + t_{dec} \quad (2-16)$$

dengan:

$t_{proc}$  = *delay* proses

$t_{enc}$  = *delay* enkapsulasi total

$t_{dec}$  = *delay* dekapsulasi total

- **Server**

Pada server terjadi proses enkapsulasi sebelum paket data ditransmisikan ke GGSN. Paket data video streaming diberi penambahan header RTP, UDP, IPv6 seperti dijelaskan sebelumnya. Apabila panjang paket video *streaming* melebihi *Maximum Transfer Unit* (MTU) *Ethernet* sebesar 1500 byte, maka paket video *streaming* akan mengalami fragmentasi menjadi beberapa buah frame sesuai dengan persamaan:

$$N_{frame\ Ethernet} = \frac{P_{VS-size}}{MTU_{ethernet}} \quad (2-17)$$

dengan:

$N_{frame\ Ethernet}$  = jumlah *frame Ethernet*

$MTU_{ethernet}$  = MTU *Ethernet* (1500 byte)

Sehingga jumlah total *frame* pada server yang dapat dikirimkan ke GGSN sesuai dengan persamaan:

$$W_{VS} = P_{VS-size} + [N_{frame\ Ethernet} \times (H_{ethernet} + FCS)] \quad (2-18)$$

Dalam skripsi ini, pada server digunakan standar *interface Gigabit Ethernet* dengan kecepatan 1 Gbps. Sehingga *delay* enkapsulasi pada server didapatkan dengan persamaan berikut:

$$t_{E1} = \frac{W_{frame\ server} - P_{VS-size}}{C_{server}} \times 8 \quad (2-19)$$



dengan:

$t_{E1}$  = delay enkapsulasi pada *server* (s)

$W_{frame\ server}$  = panjang frame pada *server* (byte)

$C_{server}$  = kecepatan pemrosesan data pada *server* (bps)

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

Pada GGSN, paket data yang diterima dari *server* mengalami proses dekapsulasi. Kemudian, sebelum paket data aplikasi video *streaming* ditransmisikan ke SGSN, dilakukan proses enkapsulasi terlebih dahulu.

Pada proses dekapsulasi, paket data aplikasi video *streaming* didapatkan sesuai dengan persamaan:

$$W_{VS\ GGSN} = W_{frame\ server} - H_{UDP} - H_{IPv6} - H_{ethernet} - FCS \quad (2-20)$$

Dengan mengasumsikan GGSN menggunakan standar *Fast Ethernet* dengan kecepatan 100 Mbps, maka besarnya *delay* dekapsulasi pada GGSN didapatkan sesuai dengan persamaan:

$$t_{D1} = \frac{W_{frame\ server} - W_{VS\ GGSN}}{C_{GGSN}} \times 8 \quad (2-21)$$

dengan:

$t_{D1}$  = delay dekapsulasi pada GGSN (s)

$W_{VS\ GGSN}$  = panjang paket data video *streaming* di GGSN (byte)

$C_{GGSN}$  = kecepatan pemrosesan data di GGSN (bps)

Besar MSS dapat diperoleh sesuai dengan persamaan berikut:

$$MSS = MTU - H_{GTP} - H_{UDP} - H_{IP} \quad (2-22)$$

Selanjutnya, paket data yang melebihi MSS akan disegmentasi menggunakan persamaan berikut

$$N_{datagram} = \frac{W_{vs-GGSN}}{MSS} \quad (2-23)$$

Kemudian, paket data aplikasi video *streaming* dienkapsulasi dengan penambahan *header* GTP (*GPRS Tunneling Protocol*), UDP (*User Datagram Protocol*), dan IP (*Internet Protocol*) sesuai dengan persamaan berikut:

$$W_{datagram} = W_{VS\ GGSN} + N_{datagram} \times (H_{GTP} + H_{UDP} + H_{IPv6}) \quad (2-24)$$

Karena panjang datagram IP melebihi *Maximum Transfer Unit* (MTU) *Ethernet* (1500 byte), maka datagram IP akan mengalami fragmentasi menjadi beberapa buah frame sesuai dengan persamaan:

$$N_{frame\ Ethernet} = \frac{W_{datagram\ IP}}{MTU_{ethernet}} \quad (2-25)$$

Sehingga, jumlah total *frame* pada GGSN yang dapat dikirimkan ke SGSN sesuai dengan persamaan:

$$W_{frame\ GGSN} = W_{datagram} + [N_{frame\ Ethernet} \times (H_{ethernet} + FCS)] \quad (2-26)$$

Sehingga *delay* enkapsulasi pada GGSN didapatkan dengan persamaan:

$$t_{E2} = \frac{W_{frame\ GGSN} - W_{VS\ GGSN}}{C_{GGSN}} \times 8 \quad (2-27)$$

dengan:

$t_{E2}$  = *delay* enkapsulasi pada GGSN (s)

$W_{frame\ GGSN}$  = panjang *frame* di GGSN (byte)

$C_{GGSN}$  = kecepatan pemrosesan data di GGSN (bps)

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

Pada SGSN, paket data yang diterima dari GGSN mengalami proses dekapsulasi. Kemudian, sebelum paket data aplikasi video *streaming* ditransmisikan ke RNC, dilakukan proses enkapsulasi terlebih dahulu.

Paket data yang diterima dari GGSN mengalami proses dekapsulasi pada SGSN sesuai dengan persamaan:

$$W_{VS\ SGSN} = W_{frame\ GGSN} - H_{GTP} - H_{UDP} - H_{IPv6} - H_{ethernet} - FCS \quad (2-28)$$

Dengan mengasumsikan bahwa SGSN menggunakan standar Fast Ethernet dengan kecepatan data 100 Mbps, maka besarnya *delay* dekapsulasi didapatkan dengan menggunakan persamaan:

$$t_{D2} = \frac{W_{frame\ GGSN} - W_{VS\ SGSN}}{C_{SGSN}} \times 8 \quad (2-29)$$

dengan:

$t_{D2}$  = *delay* dekapsulasi pada SGSN (s)

$W_{VS\ SGSN}$  = panjang paket data video *streaming* di SGSN (byte)

$Header_{GTP}$  = panjang *header* GTP (8 byte)

$C_{GGSN}$  = kecepatan pemrosesan data pada GGSN (bps)



Besar MSS dapat diperoleh sesuai dengan persamaan berikut:

$$MSS = MTU - H_{GTP} - H_{UDP} - H_{IP} \quad (2-30)$$

Selanjutnya, paket data yang melebihi MSS akan disegmentasi menggunakan persamaan berikut

$$N_{datagram} = \frac{W_{vs-GGSN}}{MSS} \quad (2-31)$$

Kemudian, paket data aplikasi video *streaming* dienkapsulasi dengan penambahan *header* GTP (*GPRS Tunneling Protocol*), UDP (*User Datagram Protocol*), IP (*Internet Protocol*), dan *header* AAL5.

$$W_{frameAAL5} = W_{VSSGSN} + [N_{datagram} \times (H_{GTP} + H_{UDP} + H_{IP} + H_{AAL5(PAD+CT)})] \quad (2-32)$$

Pada *layer* ATM, *frame* AAL5 akan disegmentasi menjadi *fixed-size cells* ATM sebesar 53byte yang terdiri dari *payload* ATM sebesar 48byte dan *header* ATM sebesar 5byte (Stallings, 2001:363)

$$N_{frameATM} = \frac{W_{frameAAL5}}{48} \quad (2-33)$$

Setelah tersegmentasi, *frame-frame* ini dienkapsulasi untuk disesuaikan dengan panjang *frame* ATM. Sehingga diperoleh panjang *frame* ATM masing-masing sesuai dengan persamaan:

$$W_{frameATM} = Payload_{ATM} + header_{ATM} \quad (2-34)$$

Setiap *frame* ATM diberi *header* 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:

$$W_{frameSGSN} = N_{frameATM} \times W_{frameATM} \quad (2-35)$$

Sehingga, *delay* enkapsulasi yang terjadi pada SGSN didapatkan dengan persamaan:

$$t_{E3} = \frac{W_{frameSGSN} - W_{VSSGSN}}{C_{SGSN-RNC}} \times 8 \quad (2-36)$$

dengan:

$t_{E3}$  = *delay* enkapsulasi pada SGSN (s)

$W_{frameSGSN}$  = panjang *frame* di SGSN (byte)

$C_{SGSN-RNC}$  = kecepatan pemrosesan data pada SGSN-RNC (bps)

- **Radio Access Network (RAN)**

Pada RAN, paket data yang diterima dari SGSN mengalami proses dekapsulasi. Kemudian, sebelum paket data aplikasi video *streaming* ditransmisikan ke *Node B*, dilakukan proses enkapsulasi terlebih dahulu.

Paket data yang diterima dari SGSN mengalami proses dekapsulasi pada RAN sesuai dengan persamaan:

$$W_{VS(RNC)} = W_{frameSGSN} - H_{GTP} - H_{UDP} - H_{IP} - H_{AAL5} - H_{ATM} \quad (2-37)$$

Pada RAN, digunakan *interface* sistem transmisi ATM STM-1 (*Synchronous Transport Module-1*) yang memberikan kecepatan data sebesar 155,52 Mbps. Nilai *delay* dekapsulasi pada RAN didapatkan dengan persamaan:

$$t_{D3} = \frac{W_{frameSGSN} - W_{VS RNC}}{C_{RNC}} \times 8 \quad (2-38)$$

dengan:

$t_{D3}$  = *delay* dekapsulasi pada RAN (s)

$W_{VS RAN}$  = panjang paket data video *streaming* pada RAN (byte)

$C_{RNC}$  = kecepatan pemrosesan data pada RAN (bps)

Pada RAN, *frame* ATM disegmentasi menjadi PDCP Data PDU (*Protocol Data Unit*) dengan menambahkan *header* PDU *type* sebesar 3 bit dan PID (*Paket Identifier*) sebesar 5 bit, jadi total 8 bit atau 1byte (3GPP TS 22.323 Rel.5)

$$W_{framePDCP} = W_{VS RAN} + header_{PDCP} \quad (2-39)$$

Selanjutnya pada layer RLC (*Radio Link Control*), *frame* PDCP disegmentasi menjadi RLC PDU *fixed size* sebesar 40 *byte*. Sehingga jumlah *frame* RLC sesuai dengan persamaan:

$$N_{frameRLC} = \frac{W_{framePDCP}}{40byte} \quad (2-40)$$

Sehingga besar paket data pada *frame* RLC dapat diketahui dengan persamaan:

$$W_{frameRLC} = 40byte + header_{RLC} \quad (2-41)$$

Setiap *frame* selanjutnya diberi *header* sebesar 2 *byte*, sehingga panjang setiap *frame* RLC sebesar 42 *byte*. Panjang *frame* RLC total yang siap



diteruskan ke *layer* MAC (*Medium Acces Control*) sesuai dengan persamaan berikut:

$$W_{frame\ RLC\ total} = N_{frame\ RLC} \times W_{frame\ RLC} \quad (2-42)$$

dengan:

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

$header_{PDCP}$  = panjang *header* PDCP (1 byte)

$N_{frame\ RLC}$  = jumlah *frame* pada RLC

$W_{frame\ RLC\ total}$  = panjang *frame* RLC total (byte)

$W_{frame\ RLC}$  = panjang *frame* RLC (42 byte)

Pada *layer* MAC, RLC PDU akan disegmentasi menjadi MAC SDU. Besarnya MAC SDU tergantung pada besarnya RLC PDU yang telah ditetapkan yaitu sebesar 42byte. Sehingga jumlah *frame* MAC SDU sesuai dengan persamaan:

$$N_{frame\ MAC} = \frac{W_{frame\ RLC\ total}}{42\text{byte}} \quad (2-43)$$

Sehingga besar paket data pada *frame* MAC dapat diketahui dengan persamaan:

$$W_{frame\ MAC} = header_{MAC} + 42\text{byte} + CRC \quad (2-44)$$

Setiap *frame* ATM diberi *header* sebesar 5 byte. Panjang *frame* yang siap ditransmisikan menuju GGSN merupakan panjang *frame* ATM total sesuai dengan persamaan:

$$W_{frame\ RNC} = N_{frame\ ATM} \times W_{frame\ ATM} \quad (2-45)$$

Sehingga, besar *delay* dekapsulasi yang terjadi pada RAN didapatkan dengan persamaan berikut:

$$t_{E4} = \frac{W_{frame\ RNC} - W_{VSRNC}}{C_{RNC}} \times 8 \quad (2-46)$$

dengan:

$t_{E4}$  = *delay* enkapsulasi pad RAN (s)

$W_{frame\ RAN}$  = panjang *frame* pada RAN (byte)

$C_{RAN}$  = kecepatan pemrosesan data pada RAN (bps)

- **User Equipment (UE)**

Pada UE, paket data yang diterima dari *Node B* mengalami proses dekapsulasi. Proses dekapsulasi pada UE sesuai dengan persamaan:

$$W_{VS\ UE} = W_{frame\ RAN} - H_{PDCP} - N_{frameMAC} \times (H_{MAC} - CRC - H_{RLC}) \quad (2-47)$$

Sesuai dengan standart 3GPP, UMTS memiliki *data rate* maksimum sebesar 384 kbps, maka besar *delay* dekapsulasi yang terjadi di UE dapat dihitung dengan persamaan:

$$t_{D(UE)} = \frac{W_{frameRAN} - W_{VS(UE)}}{C_{UE}} \times 8 \quad (2-48)$$

dengan:

$t_{D5}$  = *delay* dekapsulasi pada UE (s)

$W_{VS\ UE}$  = panjang paket data video *streaming* pada UE (byte)

$C_{UE}$  = kecepatan pemrosesan data pada UE (bps)

- **Packet Data Network Gateway (PDN-GW)**

Pada PDN-GW, paket data yang diterima dari *server* mengalami proses dekapsulasi. Kemudian, sebelum paket data aplikasi video *streaming* ditransmisikan ke S-GW, dilakukan proses enkapsulasi terlebih dahulu.

Pada proses dekapsulasi, paket data aplikasi video *streaming* didapatkan sesuai dengan persamaan:

$$W_{VS\ PDN-GW} = W_{frame\ server} - H_{UDP} - H_{IP} - H_{ethernet} - FCS \quad (2-49)$$

Dengan mengasumsikan PDN-GW menggunakan standar *Fast Ethernet* dengan kecepatan 100 Mbps, maka besarnya *delay* dekapsulasi pada PDN-GW didapatkan sesuai dengan persamaan:

$$t_{D1} = \frac{W_{frameserver} - W_{VSPDN-GW}}{C_{PDN-GW}} \times 8 \quad (2-50)$$

dengan:

$t_{D1}$  = *delay* dekapsulasi pada PDN-GW (s)

$W_{VS\ PDN-GW}$  = panjang paket data video *streaming* di PDN-GW (byte)

$C_{PDN-GW}$  = kecepatan pemrosesan data di PDN-GW (bps)

Besar MSS dapat diperoleh sesuai dengan persamaan berikut:

$$MSS = MTU - H_{GTP} - H_{UDP} - H_{IP} \quad (2-51)$$



Selanjutnya, paket data yang melebihi MSS akan disegmentasi menggunakan persamaan berikut

$$N_{\text{datagram}} = \frac{W_{\text{VSPDN-GW}}}{\text{MSS}} \quad (2-52)$$

Kemudian, paket data aplikasi video *streaming* dienkapsulasi dengan penambahan *header* GTP (GPRS Tunneling Protocol), UDP (User Datagram Protocol), dan IP (Internet Protocol) sesuai dengan persamaan berikut:

$$W_{\text{datagram}} = W_{\text{VSPDN-GW}} + [N_{\text{datagram}} \times (H_{\text{GTP}} + H_{\text{UDP}} + H_{\text{IP}})] \quad (2-53)$$

Karena panjang datagram IP melebihi *Maximum Transfer Unit* (MTU) *Ethernet* (1500 byte), maka datagram IP akan mengalami fragmentasi menjadi beberapa buah *frame* sesuai dengan persamaan:

$$N_{\text{frameEthernet}} = \frac{W_{\text{datagram}}}{\text{MTU}_{\text{ethernet}}} \quad (2-54)$$

Sehingga, jumlah total *frame* pada PDN-GW yang dapat dikirimkan ke S-GW sesuai dengan persamaan:

$$W_{\text{frame PDN-GW}} = W_{\text{datagram}} + [N_{\text{frame Ethernet}} \times (H_{\text{ethernet}} + \text{FCS})] \quad (2-55)$$

Sehingga *delay* enkapsulasi pada PDN-GW didapatkan dengan persamaan:

$$t_{E2} = \frac{W_{\text{framePDN-GW}} - W_{\text{VSPDN-GW}}}{C_{\text{PDN-GW}}} \times 8 \quad (2-56)$$

dengan:

$t_{E2}$  = *delay* enkapsulasi pada PDN-GW (s)

$W_{\text{frame PDN-GW}}$  = panjang *frame* di PDN-GW (byte)

$C_{\text{PDN-GW}}$  = kecepatan pemrosesan data di PDN-GW (bps)

- **Serving Gateway (S-GW)**

Pada S-GW, paket data yang diterima dari PDN-GW mengalami proses dekapsulasi. Kemudian, sebelum paket data aplikasi video *streaming* ditransmisikan ke eNodeB, dilakukan proses enkapsulasi terlebih dahulu.

Pada proses dekapsulasi, paket data aplikasi video *streaming* didapatkan sesuai dengan persamaan:

$$W_{\text{VSS-GW}} = W_{\text{framePDN-GW}} - H_{\text{GTP}} - H_{\text{UDP}} - H_{\text{ethernet}} - \text{FCS} \quad (2-57)$$

Dengan mengasumsikan S-GW menggunakan standar *Fast Ethernet* dengan kecepatan 100 Mbps, maka besarnya *delay* dekapsulasi pada S-GW didapatkan sesuai dengan persamaan:

$$t_{D2} = \frac{W_{framePDN-GW} - W_{VSS-GW}}{C_{S-GW}} \times 8 \quad (2-58)$$

dengan:

$t_{D2}$  = *delay* dekapsulasi pada S-GW (s)

$W_{VSS-GW}$  = panjang paket data video *streaming* di S-GW (byte)

$C_{S-GW}$  = kecepatan pemrosesan data di S-GW (bps)

Besar MSS dapat diperoleh sesuai dengan persamaan berikut:

$$MSS = MTU - H_{GTP} - H_{UDP} - H_{IP} \quad (2-59)$$

Selanjutnya, paket data yang melebihi MSS akan disegmentasi menggunakan persamaan berikut

$$N_{datagram} = \frac{W_{VSS-GW}}{MSS} \quad (2-60)$$

Kemudian, paket data aplikasi video *streaming* dienkapsulasi dengan penambahan *header* GTP (*GPRS Tunneling Protocol*), UDP (*User Datagram Protocol*), dan IP (*Internet Protocol*) sesuai dengan persamaan berikut:

$$W_{datagram} = W_{VSS-GW} + [N_{datagram} \times (H_{GTP} + H_{UDP} + H_{IP})] \quad (2-61)$$

Karena panjang datagram IP melebihi *Maximum Transfer Unit* (MTU) *Ethernet* (1500 byte), maka datagram IP akan mengalami fragmentasi menjadi beberapa buah frame sesuai dengan persamaan:

$$N_{frameEthernet} = \frac{W_{datagram}}{MTU_{ethernet}} \quad (2-62)$$

Sehingga jumlah total *frame* pada S-GW yang dapat dikirimkan ke eNodeB sesuai dengan persamaan:

$$W_{frame\ S-GW} = W_{datagram} + [N_{frame\ Ethernet} \times (H_{ethernet} + FCS)] \quad (2-63)$$

Sehingga *delay* enkapsulasi pada S-GW didapatkan dengan persamaan:



$$t_{E3} = \frac{W_{frameS-GW} - W_{VSS-GW}}{C_{S-GW}} \times 8 \quad (2-64)$$

dengan:

$t_{E3}$  = *delay* enkapsulasi pada S-GW (s)

$W_{frame S-GW}$  = panjang *frame* di S-GW (byte)

$C_{S-GW}$  = kecepatan pemrosesan data di S-GW (bps)

- **Evolved Node B (eNodeB)**

Pada eNodeB, paket data yang diterima dari S-GW mengalami proses dekapsulasi. Kemudian, sebelum paket data aplikasi video *streaming* ditransmisikan ke UE, dilakukan proses enkapsulasi terlebih dahulu.

Pada proses dekapsulasi, paket data aplikasi video *streaming* didapatkan sesuai dengan persamaan:

$$W_{VS eNodeB} = W_{frame S-GW} - H_{GTP} - H_{UDP} - H_{IP} - H_{ethernet} - FCS \quad (2-65)$$

eNodeB menggunakan *interface* STM-1 (*Synchronous Transport Module-1*) dengan kecepatan 155,52 Mbps. Maka besarnya *delay* dekapsulasi didapatkan dengan menggunakan persamaan:

$$t_{D3} = \frac{W_{frameS-GW} - W_{VSeNodeB}}{C_{eNodeB}} \times 8 \quad (2-66)$$

dengan:

$t_{D3}$  = *delay* dekapsulasi pada eNodeB (s)

$W_{VS eNodeB}$  = panjang paket data video *streaming* di eNodeB (byte)

$C_{eNodeB}$  = kecepatan pemrosesan data di eNodeB (bps)

Pada *layer* PDCP (*Packet Data Convergence Protocol*), paket datagram IP video *streaming* dienkapsulasi dengan penambahan *header* sebesar 1 byte, yaitu *header* PDU. Banyaknya paket IP pada eNodeB dapat diperoleh sebagai berikut:

$$N_{datagram} = \frac{W_{VSeNodeB}}{MTU} \quad (2-67)$$

Pada *layer* PDCP, panjang *frame* dapat dihitung dengan menggunakan persamaan:

$$W_{frame PDCP} = W_{VS eNodeB} + H_{PDCP} \quad (2-68)$$

Selanjutnya pada *layer* RLC (*Radio Link Control*), *frame* PDCP disegmentasi menjadi RLC PDU *fixed size* sebesar 40 *byte*. Sehingga jumlah *frame* RLC sesuai dengan persamaan

$$N_{frameRLC} = \frac{W_{framePDCP}}{40byte} \quad (2-69)$$

dengan:

$$\begin{aligned} N_{frameRLC} &= \text{jumlah } frame \text{ RLC} \\ W_{framePDCP} &= \text{panjang } frame \text{ PDCP (byte)} \end{aligned}$$

Setiap *frame* selanjutnya diberi *header* sebesar 2 *byte*, sehingga panjang setiap *frame* RLC sebesar 42 *byte*. Panjang *frame* RLC total yang siap diteruskan ke *layer* MAC (*Medium Access Control*) sesuai dengan persamaan:

$$W_{frameRLCtotal} = N_{frameRLC} \times W_{frameRLC} \quad (2-70)$$

Pada *layer* MAC, RLC disegmentasi menjadi MAC SDU (*Service Data Unit*) sebesar 42 *byte*. Jumlah *frame* SDU sesuai dengan persamaan:

$$N_{frameMAC} = \frac{W_{frameRLCtotal}}{42byte} \quad (2-71)$$

dengan:

$$\begin{aligned} N_{frameMAC} &= \text{jumlah } frame \text{ MAC} \\ W_{frameRLCtotal} &= \text{panjang } frame \text{ RLC total (byte)} \end{aligned}$$

Sedangkan panjang *frame* MAC dihitung dengan menggunakan persamaan:

$$W_{frameMAC} = H_{MAC} + 42 \text{ byte} \quad (2-72)$$

Panjang *frame* yang siap ditransmisikan menuju UE merupakan panjang *frame* MAC total sesuai dengan persamaan:

$$W_{frameeNodeB} = N_{frameMAC} \times W_{frameMAC} \quad (2-73)$$

Sehingga besar *delay* dekapsulasi yang terjadi pada eNodeB didapatkan dengan menggunakan persamaan:

$$t_{EA} = \frac{W_{frameeNodeB} - W_{VSeNodeB}}{C_{eNodeB}} \times 8 \quad (2-74)$$



dengan:

$t_{E4}$  = *delay* enkapsulasi pada eNodeB (s)

$W_{VS\ eNodeB}$  = panjang paket data video *streaming* di eNodeB (byte)

$C_{eNodeB}$  = kecepatan pemrosesan data di eNodeB (bps)

- **User Equipment (UE)**

Pada UE, paket data yang diterima dari eNodeB mengalami proses dekapsulasi. Proses dekapsulasi pada UE sesuai dengan persamaan:

$$W_{VS\ UE} = W_{frame\ eNodeB} - H_{GTP} - H_{UDP} - H_{IP} - H_{PDCP} - H_{RLC} - H_{MAC} \quad (2-75)$$

Dengan mengasumsikan *data rate* maksimum sebesar 3,6 Mbps, maka besarnya *delay* dekapsulasi yang terjadi di UE didapatkan dengan persamaan:

$$t_{D4} = \frac{W_{frame\ eNodeB} - W_{VS\ UE}}{C_{UE}} \times 8 \quad (2-76)$$

dengan:

$t_{D4}$  = *delay* dekapsulasi pada UE (s)

$W_{VS\ UE}$  = panjang paket data video *streaming* pada UE (byte)

$C_{UE}$  = kecepatan pemrosesan data pada UE (bps)

### 2.5.2.3 Delay Transmisi

*Delay* transmisi adalah waktu yang dibutuhkan untuk meletakkan sebuah paket data ke media transmisi. Dipengaruhi ukuran paket data dan kecepatan transmisi.

*Delay* transmisi dapat ditentukan dengan persamaan [Mischa Schwartz, 1987 : 132]:

$$t_T = \frac{W}{C} \quad (2-77)$$

Sehingga *delay* transmisi total pada UMTS ditentukan dengan persamaan:

$$t_{T\ tot} = t_{T1} + t_{T2} + t_{T3} + t_{T4} \quad (2-78)$$

dimana:

$t_{T\ tot}$  = *delay* transmisi total (s)

$W$  = panjang *frame* pada *node* (bit)

$C$  = kecepatan transmisi (bps)

- $t_{T1}$  = *delay* transmisi pada GGSN-SGSN (s)  
 $t_{T2}$  = *delay* transmisi pada SGSN-RNC (s)  
 $t_{T3}$  = *delay* transmisi pada RNC-Node B(s)  
 $t_{T4}$  = *delay* transmisi pada Node B-UE (s)

Sehingga *delay* transmisi total pada LTE ditentukan dengan persamaan:

$$t_{T \text{ tot}} = t_{T1} + t_{T2} + t_{T3} + t_{T4} \quad (2-79)$$

dimana:

- $t_{T \text{ tot}}$  = *delay* transmisi total (s)  
 $W$  = panjang *frame* pada *node* (bit)  
 $C$  = kecepatan transmisi (bps)  
 $t_{T1}$  = *delay* transmisi pada *server* – PDN-GW (s)  
 $t_{T2}$  = *delay* transmisi pada PDN-GW – S-GW (s)  
 $t_{T3}$  = *delay* transmisi pada S-GW – eNodeB (s)  
 $t_{T4}$  = *delay* transmisi pada eNodeB -UE (s)

#### 2.5.2.4 Delay Propagasi

*Delay* propagasi adalah waktu yang dibutuhkan untuk merambatkan paket data melalui media transmisi dari SGSN ke UE. *Delay* propagasi ditentukan dengan persamaan [Forouzan, Behrouz A, 2000 : 215]:

$$t_p = \frac{N_{\text{frame}} \times R}{v} \quad (2-80)$$

Sehingga *delay* propagasi total UMTS ditentukan dengan persamaan:

$$t_{p \text{ tot}} = t_{p1} + t_{p2} + t_{p3} + t_{p4} \quad (2-81)$$

dimana:

- $t_{p \text{ tot}}$  = *delay* propagasi (s)  
 $N_{\text{frame}}$  = jumlah *frame* pada *node*  
 $R$  = jarak antar *node* (m)  
 $v$  = cepat rambat gelombang =  $2 \times 10^8$  (m/s)  
 $t_{p1}$  = *delay* propagasi pada GGSN-SGSN (s)  
 $t_{p2}$  = *delay* propagasi pada SGSN-RNC (s)  
 $t_{p3}$  = *delay* propagasi pada RNC-BTS(s)  
 $t_{p4}$  = *delay* propagasi pada BTS-UE (s)



Sehingga *delay* propagasi total LTE ditentukan dengan persamaan:

$$t_{p \text{ tot}} = t_{p1} + t_{p2} + t_{p3} + t_{p4} \quad (2-82)$$

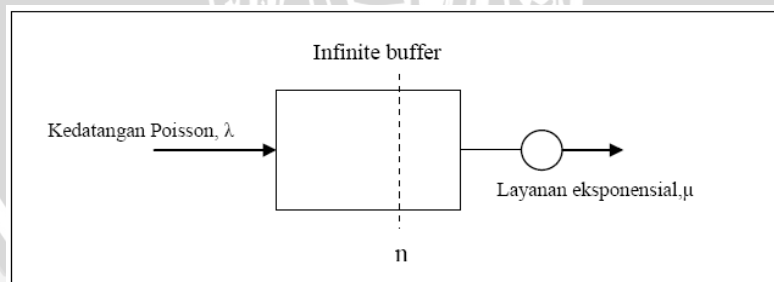
dimana:

- $t_{p \text{ tot}}$  = *delay* propagasi (s)
- $N_{\text{frame}}$  = jumlah *frame* pada node
- $R$  = jarak antar *node* (m)
- $v$  = cepat rambat gelombang =  $3 \times 10^8$  (m/s)
- $t_{p1}$  = *delay* propagasi pada *server* – PDN-GW (s)
- $t_{p2}$  = *delay* propagasi pada PDN-GW – S-GW (s)
- $t_{p3}$  = *delay* propagasi pada S-GW - eNodeB (s)
- $t_{p4}$  = *delay* propagasi pada eNodeB -UE (s)

### 2.5.2.5 Delay Antrian

*Delay* antrian adalah waktu dimana paket data berada dalam antrian untuk ditransmisikan. Selama waktu ini, paket data menunggu hingga paket yang lain selesai ditransmisikan. *Delay* antrian dapat dihitung dengan menggunakan model antrian M/M/1. M pertama menunjukkan kedatangan *Poisson*, M kedua berarti distribusi waktu pelayanan eksponensial, dan 1 menunjukkan 1 menunjukkan jumlah server yang akan melayani pelanggan.

Disiplin antrian yang digunakan adalah FIFO (*First In First Out*). Bentuk model antrian M/M/1 dapat dilihat pada Gambar 2.15.



**Gambar 2.15** Model antrian M/M/1  
(Sumber: Mischa Schwartz, 1987:31)

*Delay* antrian terjadi pada setiap *node* pada jaringan LTE, yaitu PDN-GW, S-GW, dan eNodeB. Pada jaringan UMTS, yaitu GGSN, SGSN, RNC, dan NodeB. Besarnya *delay* antrian pada *node* ditentukan dengan persamaan (Mischa Schwartz, 1987 : 42):

$$t_w = t_{queue} + t_{serv} \quad (2-83)$$

dengan:

$t_w$  = *delay* antrian (s)

$t_{queue}$  = waktu tunggu paket pada *node* (s)

$t_{serv}$  = waktu rata-rata pelayanan *node* (s)

Perhitungan waktu rata-rata pelayanan antrian data di *node* ( $t_{serv}$ ) dihitung dengan menggunakan persamaan (Mischa Schwartz, 1987 : 23):

$$t_{serv} = \frac{L}{C} \quad (2-84)$$

dengan:

$\mu$  = kecepatan pelayanan *node* (bps)

$L$  = panjang paket data di *node* (bit)

$C$  = kecepatan transmisi pada *node* (bps)

Sedangkan waktu tunggu paket pada *router* ( $t_{queue}$ ) dihitung dengan menggunakan persamaan (Mischa Schwartz, 1987 : 42):

$$t_{queue} = \frac{\lambda}{\mu^2(1-\rho)} \quad (2-85)$$

Untuk performansi sistem antrian, ditunjukkan dalam bentuk  $\rho$  (faktor utilisasi), yang nilainya diasumsikan berubah-ubah dengan kenaikan tertentu. Faktor utilisasi bernilai lebih besar dari 0 dan lebih kecil dari 1 dengan kenaikan sebesar 0,1. Hal ini disebabkan karena jika  $\rho > 1$  berarti rata-rata kedatangan melampaui rata-rata pelayanan atau server tidak dapat menjaga rata-rata kedatangan dan panjang antrian yang bertambah tanpa batas. Besarnya faktor utilisasi dapat dihitung dengan menggunakan persamaan:

$$\rho = \frac{\lambda}{\mu} \quad \Leftrightarrow \quad \lambda = \mu\rho \quad (2-86)$$

Sehingga dari persamaan 2-85 dan 2-86 *delay* antrian dapat ditentukan dengan persamaan:

$$t_w = \frac{\lambda}{\mu(\mu-\lambda)} + \frac{1}{\mu} \quad (2-87)$$

dengan:

$t_{queue}$  = waktu tunggu paket pada *node* (s)

$\lambda$  = kecepatan kedatangan paket pada *node* (bps)



$\rho$  = faktor utilitas ( $0 < \rho < 1$ )

$\mu$  = kecepatan pelayanan *node* (bps)

Sehingga *delay* antrian total LTE ditentukan dengan persamaan:

$$t_{w \text{ tot}} = t_{w1} + t_{w2} + t_{w3} \quad (2-88)$$

dengan:

$t_{w \text{ tot}}$  = *delay* antrian total (s)

$t_{w1}$  = *delay* antrian pada PDN-GW (s)

$t_{w2}$  = *delay* antrian pada S-GW (s)

$t_{w3}$  = *delay* antrian pada eNodeB (s)

Sehingga *delay* antrian total UMTS ditentukan dengan persamaan:

$$t_{w \text{ tot}} = t_{w1} + t_{w2} + t_{w3} + t_{w4} \quad (2-89)$$

dengan:

$t_{w \text{ tot}}$  = *delay* antrian total (s)

$t_{w1}$  = *delay* antrian pada GGSN (s)

$t_{w2}$  = *delay* antrian pada SGSN (s)

$t_{w3}$  = *delay* antrian pada RNC (s)

$t_{w4}$  = *delay* antrian pada NodeB (s)

### 2.5.3 Throughput

*Throughput* merupakan parameter yang digunakan untuk mengetahui jumlah paket yang diterima dalam keadaan baik terhadap waktu total transmisi yang dibutuhkan dari sumber data ke penerima. Pengiriman data pada jaringan *packet switching* antara 2 stasiun yang melalui beberapa lapisan protokol mempunyai batas kemampuan kapasitas yang bisa dikeluarkan oleh jaringan tersebut. Pada keadaan saturasi, yaitu suatu kondisi dimana selalu tersedia *frame* yang menunggu untuk ditransmisikan,  $t_v$  merepresentasikan sebagai waktu transmisi rata-rata *frame* yang diterima dengan benar (Mischa Schwartz, 1987 : 129).

$$\lambda = \frac{\text{jumlah paket data yang diterima}}{t_v} \quad (\text{paket/s}) \quad (2-90)$$

Dengan:

$\lambda$  = *throughput* (paket/s)

$t_v$  = waktu total transmisi untuk mengirimkan paket yang benar (s)

*Throughput* yang didapat dengan memperhitungkan probabilitas *packet loss* ( $\rho$ ) adalah (Mischa Schwartz, 1987 : 129):

$$\lambda = \frac{1}{t_v} = \frac{(1-\rho_{tot})}{t_1[1+(\alpha-1)\rho_{tot}]} \quad (\text{paket/s}) \quad (2-91)$$

dan,

$$\alpha = 1 + \frac{t_{out}}{t_1}, \quad \text{dimana } t_{out} = 2t_p + 2t_1 + 2t_{proc} \quad (2-92)$$

Dengan:

$\lambda$  = *throughput* (paket/s)

$\rho_{tot}$  = probabilitas *frame error* total yang terjadi dalam jaringan

$t_{out}$  = interval *time out* (s)

$t_{proc}$  = *delay* proses, *delay* enkapsulasi-dekapsulasi dan antrian (s)

$t_1$  = waktu transmisi sebuah paket data *frame* dalam jaringan (s)

*Delay* propagasi total dihitung dengan cara:

$$t_p = \frac{\sum_{n=CN}^{MT} t_{pn}}{N_{frame}} \quad (\text{s}) \quad (2-93)$$

Dengan:

$t_p$  = *delay* propagasi total (s)

$\sum_{n=CN}^{MT} t_{pn}$  = jumlah *delay* propagasi dari *node* CN sampai *node* MT (s)

$N_{frame}$  = jumlah *frame* pada proses segmentasi

*Delay* proses total dihitung dengan cara:

$$t_{proc} = \frac{\sum_{n=CN}^{MT} (t_{DEn} + t_{wn})}{N_{frame}} \quad (\text{s}) \quad (2-94)$$

Dengan:

$t_{proc}$  = *delay* proses total (s)

$\sum_{n=CN}^{MT} t_{pn}$  = jumlah *delay* propagasi dari *node* CN sampai *node* MT (s)

$N_{frame}$  = jumlah *frame* pada proses segmentasi

$\sum_{n=CN}^{MT} t_{DEn}$  = jumlah *delay* dekapsulasi dan enkapsulasi dari *node* CN sampai *node* MT (s)

$\sum_{n=CN}^{MT} t_{wn}$  = jumlah *delay* antrian dari *node* CN sampai *node* MT (s)

$N_{frame}$  = jumlah *frame* pada proses segmentasi



Waktu transmisi sebuah paket data *frame* dalam jaringan mempunyai persamaan berikut (Mischa Schwartz, 1987 : 132) :

$$t_1 = \frac{\text{payload}_{frame} + H_{frame} \times 8}{C_{trans}} \quad (\text{s}) \quad (2-95)$$

Dengan:

$t_1$  = waktu transmisi sebuah paket data *frame* dalam jaringan (s)

$\text{payload}_{frame}$  = kapasitas frame (bit)

$H_{frame}$  = header frame (bit)

$C_{trans}$  = kecepatan transmisi (bps)

Probabilitas *frame error* yang terjadi dalam pentransmisian data meliputi probabilitas *frame error* dalam *application layer*, *transport layer*, dan *network layer*, serta probabilitas *frame error* yang terjadi dalam jaringan itu sendiri, sehingga dapat dirumuskan:

$$\rho = 1 - [(1 - \rho_{network})(1 - \rho_{TCP-UDP/IP})] \quad (2-96)$$

Dengan probabilitas *frame error* yang terjadi pada *application layer*, *transport layer*, dan *network layer* dapat dituliskan sebagai berikut:

$$\rho_{TCP-UDP/IP} = P_{size} \cdot \rho_b \quad (2-97)$$

Probabilitas *frame error* yang terjadi dalam jaringan merupakan gabungan dari probabilitas *frame error* yang terjadi di *airlink* dan probabilitas *error* di *IP backbone*, yang dapat dirumuskan sebagai berikut:

$$\rho_{network} = 1 - [(1 - \rho_{AI})(1 - \rho_{IP backbone})^h] \quad (2-98)$$

Dimana peobabilitas *frame error* *IP backbone* adalah :

$$\rho_{IP backbone} = (l - l')P_b \quad (2-99)$$

Sedangkan probabilitas *frame error* pada *air interface* dirumuskan :

$$\rho_{AI} = 1 - (l + P_e)^{l+l'} \quad (2-100)$$

Dengan :

$\rho_{TCP-UDP/IP}$  = probabilitas *frame error* dalam *application layer*, *transport layer*, dan *network layer*

- $n$  = jumlah *node* dalam jaringan dengan proses enkapsulasi TCP/IP
- $\rho_b$  = probabilitas *blocking*
- $P_e$  = probabilitas *bit error* (BER)
- $\rho_{network}$  = probabilitas *frame error* dalam jaringan
- $\rho_{IP backbone}$  = probabilitas *frame error* pada *IP backbone*
- $h$  = jumlah *hop* yang dilalui
- $\rho_{AI}$  = probabilitas *frame error* pada *air interface*
- $l$  = panjang data/paket (byte)
- $l'$  = panjang *header*/paket (byte)

UNIVERSITAS BRAWIJAYA

