BAB II
TINJAUAN PUSTAKA

2.1 Umum

Layanan VoD dapat dinikmati setiap penggunanya baik pengguna *mobile phone* berbasis OS maupun pengguna laptop. VoD membutuhkan *bandwidth* dan kecepatan untuk *streaming* relatif lebar dan cepat. *Long term Evolution* (LTE) mampu memberikan kecepatan dan kemampuan transfer data hingga mencapai 100Mbps dari sisi *downlink* dan 50Mbps dari sisi *uplink* yang baru tersebar di Kota Malang. Jaringan LTE mampu mencukupi kebutuhan *bandwidth* dan kecepatan untuk layanan VoD. Namun pada saat ini meskipun jaringan LTE sudah tersedia, layanan VoD untuk seluler belum sering muncul. Oleh karena itu perlu dilaksanakan penelitian untuk layanan VoD dan dilewatkan pada jaringan LTE. Kemudian dilaksanakan pengujian kinerja jaringan LTE yang dilewati berdasarkan parameter QoS jaringan dan perhitungan teoritis agar menjadi acuan bagi bahan penelitian dan pengembangan berikutnya untuk pengadaan layanan VoD.

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 Format Video

Video format mencakup dua perihal yang berbeda, baik secara teknologi maupun konsep: *Containers* (kadang disebut sebagai *Wrappers*) dan *Codecs* (singkatan *codec*/*decoder*). *Codec* digunakan dalam sebuah *container*.

Container mendeskripsikan struktur dari sebuah *file*, yakni dimana beberapa bagian yang disimpan, bagaimana mereka disisipkan, dan *codec* apa yang digunakan pada setiap bagian. *Container* juga mendeskripsikan *codec* suara sebagaimana *codec* video yang digunakan. *Container* digunakan untuk mengemas video dan komponennya (audio/metadata) dan teridentifikasi dengan ekstensi *file*, seperti .AVI, .MP4 atau .MOV

Tabel 2.1 Daftar *Container Codec*

Jenis Container	Keterangan
AVI (*Audio Video Interleave*)	Standar *container* untuk *file multimedia* dengan sistem operasi Windows
MP4 (dikenal sebagai .mp4)	Bentuk *container* standar untuk MPEG-4
FLV (*Flash Video*)	Format yang digunakan untuk mengirimkan video MPEG melewati *Flash Player*
MOV	Format *container* untuk program Quicktime milik Apple
OGG, OGM dan OGV	*Container* dengan standar *opensource*
MKV (Mastroska)	*Container* lain dengan standar *opensource* yang biasa digunakan untuk anime
VOB (DVD *Video Object*)	*Container* 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 sebuah cara mengenkodekan suara atau video menjadi aliran bit-bit data (*bytes*). Ini adalah metode yang digunakan untuk mengenkodekan video dan faktor penentu utama dari kualitas video. Codec terdiri dari *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 dikodekan dan memaikannya di sisi *client*.

Terdapat 2 jenis *Codec* yang dapat digunakan dalam *live 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
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* dilakukan *encode-encode* berulang kali sehingga jika menggunakan *lossy codec* kualitas video akan jauh menurun di bandingankan dengan video aslinya

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	*Video Conference, low delay*
H.263	1992	28,8 k-768 k	128 x 96 to 720 x 480	*Video Conference*
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 *mobile web*
AVC, H.264/ MPEG-4 part 2	14496-10 2002	100 K – 500 K	144p, 240p, 360p, 480p, 360p, 720p, 1080p	Keperluan umum, *Video On Demand*

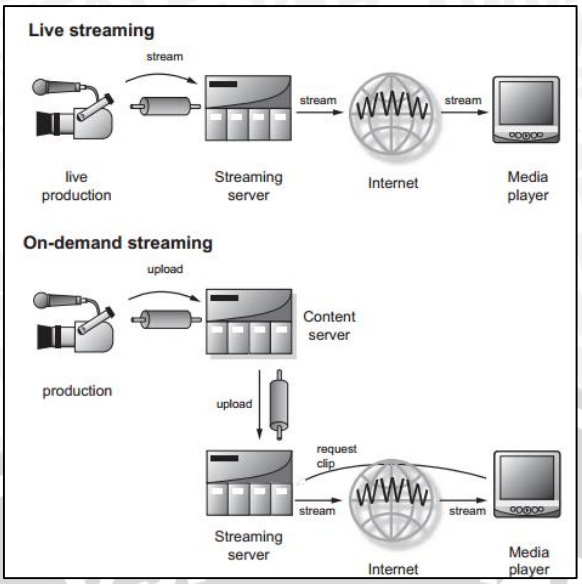
Sumber: Austerberry, 2005: 89

2.2.2 Live Streaming

Streaming adalah sebuah jenis layanan yang langsung mengolah data yang diterima tanpa menunggu seluruh data selesai terkirim. Layanan yang bersifat streaming saat ini adalah audio dan video.

Teknologi *streaming* dikenal juga sebagai *streaming* media yaitu suatu teknologi untuk memainkan atau menjalankan *file* (audio maupun video) dari sebuah *server streaming* (*web page)* baik secara langsung maupun secara rekaman, dimana data tersebut harus dikodekan terlebih dahulu menggunakan data rate tertentu yang cocok untuk ditransmisikan melalui internet atau jaringan yang sesuai dengan kapasitas *bandwidth* dari pengguna. *Data rate* yang digunakan harus cocok dengan data rate pelanggan. Untuk itu harus dilakukan encoding file audio maupun video dengan bermacam – macam dengan kecepatan *date rate* kemudian pengguna dapat menyesuaikan dengan kecepatan jaringannya maupun kecepatan dari sistem akses datanya.

Live streaming adalah tayangan langsung yang disiarkan kepada banyak dalam waktu yang bersamaan dengan kejadian aslinya, melalui media data komunikasi (network*)* baik yang terhubung kabel atau wireless

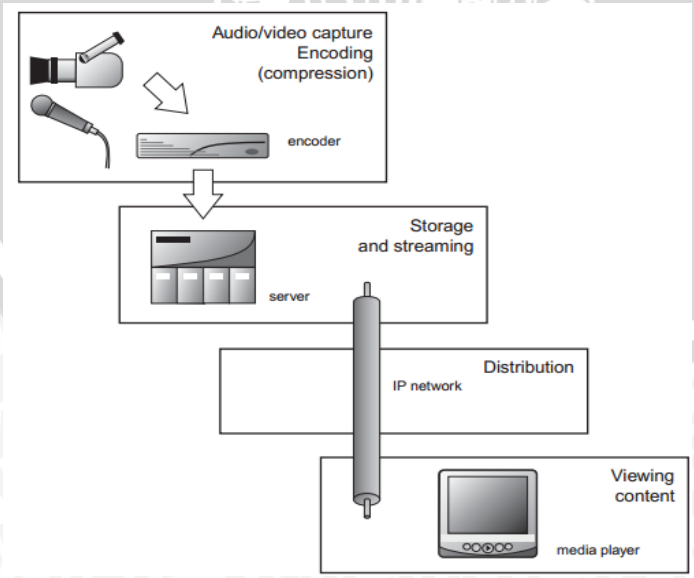


Gambar 2.1 *Live Streaming* dan *On-Demand*
Sumber: Austerberry, 2005: 144

Ada dua jenis tipe *video streaming*, yaitu:

1. *Live Streaming*, dimana tayangan yang ditampilkan merupakan siaran langsung.
2. VoD (*Video On Demand*) dimana *video* yang ditampilkan sudah terlebih dahulu direkam (*pre-encoded*) atau disimpan dalam *server* seperti yang ditunjukkan pada gambar 2.2 di atas.

Faktor-faktor yang berpengaruh dalam distribusi *video streaming* melalui jaringan antara lain besar *bandwidth* tersedia yang bervariasi (terhadap waktu), *delay* (waktu tunda), *lost packet*, dan juga teknik mendistribusikan *video* tersebut ke beberapa tujuan secara merata dan efisien. (Apostolopoulos, 2002: 1)



Gambar 2.2 Arsitektur *Streaming*
Sumber: Austerberry, 2005: 139

Pada gambar 2.2 ditunjukkan bahwa dalam arsitektur *streaming* terdapat empat buah komponen, yakni

Tabel 2.3 Komponen *Streaming*

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

Sumber: Austerberry, 2015: 145

2.2.3 Protokol *Video Streaming*

Beberapa antenna yang digunakan untuk melalakukan *video streaming* atau *data streaming* adalah sebagai berikut (Anton, hal 10):

- *User Data Protocol* (UDP)

Merupakan salah satu antenna yang digunakan dalam jaringan yang digunakan untuk mengalirkan data secara terus menerus, digunakan UDP karena dengan antennaa ini tidak memerkukan mekanisme reliabilitas, dalam arti banyaknya data yang dikirimkan tidak perlu diperhatikan jumlah paket yang hilang, hal ini bertujuan agar paket data yang dikirimkan lebih cepat, dan tidak ada mekanisme pengiriman ulang

- *Real Time Transport Protocol* (RTP)

Merupakan suatu standar untuk mengirimkan data multimedia secara *real time* yang terjadi dalam jaringan. Protokol RTP ini bergantung pada protocol *transport*, penggunaan RTP biasa banyak banyak terjadi di UDP

- *Real Time Control Protocol* (RTCP)

Merupakan sebagai QOS (Quality *of Service)* yang digunakan untuk menjamin sebuah kualitas streaming. RTCP merupakan bagian yang digunakan untuk melakukan pengontrolan terhadap paket data yang ada pada RTP

- *Real Time Streaming Protocol* (RTSP)

Merupakan protocol yang digunakan oleh program streaming multimedia untuk mengatur pengiriman data secara *real time*, tidak tergantung pada antennaa *transport.* Metode yang ada pada RTSP yaitu *play, setup, record, pause,* dan *teardown.*

2.2.4 Karakteristik Video *Streaming*

Karakteristik dari aplikasi *streaming* adalah sebagai berikut:

- Distribusi data berupa audio, video dan multimedia pada jaringan secara *real time, live casting,* atau video on demand.
- *Transfer media* data digital oleh *server* dan diterima oleh client sebagai *real time stream simultan.*
- *Client* tidak perlu menunggu keselurahan data di *download* karena server mengirimkan data yang diperlukan setiap selang waktu tertentu.
- Pada aplikasi *streaming* melibatkan jaringan , interaksi *client* dan *server*

Jenis aplikasi komunikasi *Video Broadcast*: (Kahfi, "Live Streaming" para.5) *Broadcast* bersifat *One to many*, pengirim menggunakan *channel* yang berbeda untuk setiap penerima seperti *Digital Video Broadcast for Handled* (DVB-H). *Multicast* bersifat *One to many* hanya pada client tertentu seperti *IP-Multicast over the Internet, Multimedia Broadcast Multicast Service* (MBMS). *Unicast / point-to-point* bersifat *One to one* dengan *property* tergantung pada *available back channel* yaitu dengan *back channel* atau tidak. Contoh dari *unicast* yaitu *video phone, unicast over the internet, video on demand (*VoD*)* dan *live streaming.*

Kualitas *video digital* dapat dipengaruhi 3 faktor: (Kahfi "Video dan Audio Digital "para.1)

- *Frame rate*

Merupakan banyak gambar ditampilkan / detik pada sisi *receiver* (*frame persecond*). *Frame rate* digunakan untuk sinkronisasi gambar, suara, dan kecepatanya tergantung pada kecapatan koneksi internet. Standar *National Television Standards Comitte* (NTSC) untuk

full motion video adalah 30 fps, program TV memiliki *frame rate* 24 fps dan *video digital* pada internet memilika *frame rate* 15-30 fps.

- *Color Depth*

Yaitu banyak bit / *pixel* sebagai representasi warna, contoh 24 bit merepresentasi 16,7 juta warna, 16 bit dengan 65.536 warna dan 8 bit hanya 256 warna

- *Frame Resolution*

Yaitu besaran lebar dan tinggi dalam *pixel,* contoh *full screen PC display* memiliki resolusi 640x480

- *Bit rate*

Dapat ditentukan dengan strategi berikut:

a. *Constant Bit Rate / CBR* adalah strategi memaksimalkan kualitas visual video dengan mengkonstankan *bit rate / frame, buffer* dan *feedback* pada *encoding.* Strategi ini digunakan pada *real time* dan *non-buffered video streaming* karena ketersediaan *bandwidth* tetap namun kapasitas channel terbatas, contoh *video conference* pada *channel* dengan *fixed bandwidth*

b. *Variable Bit Rate / VBR* adalah strategi memaksimalkan kualitas visual video dengan meminimalisir *bit rate,* karena hal ini berbanding lurus dengan distorsi yang terjadi pada setiap frame. Pada *fast motion scene,* VBR menggunakan lebih banyak bit dari pada *slow motion scene,* pada durasi yang sama untuk mengkosistenkan kualitas visul seperti DVD

Seperti video, juga terdapat beberapa variable yang mempengaruhi ukuran kompresi audio yaitu: (Kahfi “Video dan Audio Digital” para.4)

- *Sample Rate,* yaitu frekuensi bit dara yang diambil secara acak (*Kilohertz* /kHz). Nilai umum untuk variable ini adalah 11.024 kHz, 22.050 kHz, 44.1 kHz, 48 kHz. Makin tinggi *sample rate* maka makin baik kualitas audio, namun makin sulit mengatur sinkronisasi *audio* dan *video.*
- *Sample Size,* yaitu banyak bit yang merepresentasikan amplitude sampe audio. Nilai umum untuk *audio codec* menggunakan 16 bit – 8 bit *sample size. Channel*, yaitu banyak jenis sinyal audio, *mono* atau *stereo.*

2.2.5 Kompresi Video

Pada video untuk memperoleh kualitas video dan audi maka diperlukan Sistem Kompresi (*Encoding System*) yaitu proses mengurangi redudansi yang ada pada data.

Untuk *Encoding Video Streaming* terbagi atas *Encoding Video* dan *Enconding Audio.* Sistem kompresi dipengaruhi oleh *Video codec, Audio codec, bit rate video,* dan *audio, Frame rate video, frequency audio,* dan *size video.* Kompresi video dilakukan dengan mengurangi terjadinya redudansi pada video signal

Pada kompresi video terdapat 3 pendekatan untuk mengurangi redudansi pada data berdasarkan identifikasi tipenya: (Kahfi. "Kompresi Video" para 4)

- *Spatial redundancy,* mendefenisikan redudansi data pada video berdasarkan nilai pixel yang berkolerasi dengan tetangganya.
- *Temporal redundancy,* mendefinisikan redudansi data pada video berdasarkan nilai pixel yang berkolerasi dengan frame setelahnya.
- *Spectral redundancy,* mendefinisikan redudansi data pada video berdasarkan kolerasi 11ntenn komponen warna pada image yang sama

2.3 *LTE (Long Term Evolution)*

LTE adalah standar terbaru dalam teknologi jaringan seluler dibandingkan GSM/EDGE dan UMTS/HSPA. LTE adalah sebuah nama baru dari layanan yang mempunyai kemampuan tinggi dalam sistem komunikasi bergerak yang merupakan langkah menuju generasi ke-4 (4G) dari teknologi radio yang dirancang untuk meningkatkan kapasitas dan kecepatan jaringan telepon *mobile*. LTE adalah suatu proyek dalam *third generation partnership project* (3GPP).

LTE merupakan evolusi dari jaringan seluler yang dipersiapkan untuk teknologi 4G. Adapun tujuan pengembangan teknologi pada 3GPP adalah sebagai berikut:

1. Kebutuhan akan pengembangan jaringan 3G dalam waktu yang akan datang.
2. Kebutuhan pelangga akan kecepatan data yang tinggi dan *quality of service* (QOS).
3. Pengembangan teknologi *packet switching.*
4. Mengurangi biaya operasional karena arsitektur jaringan yang sederhana.

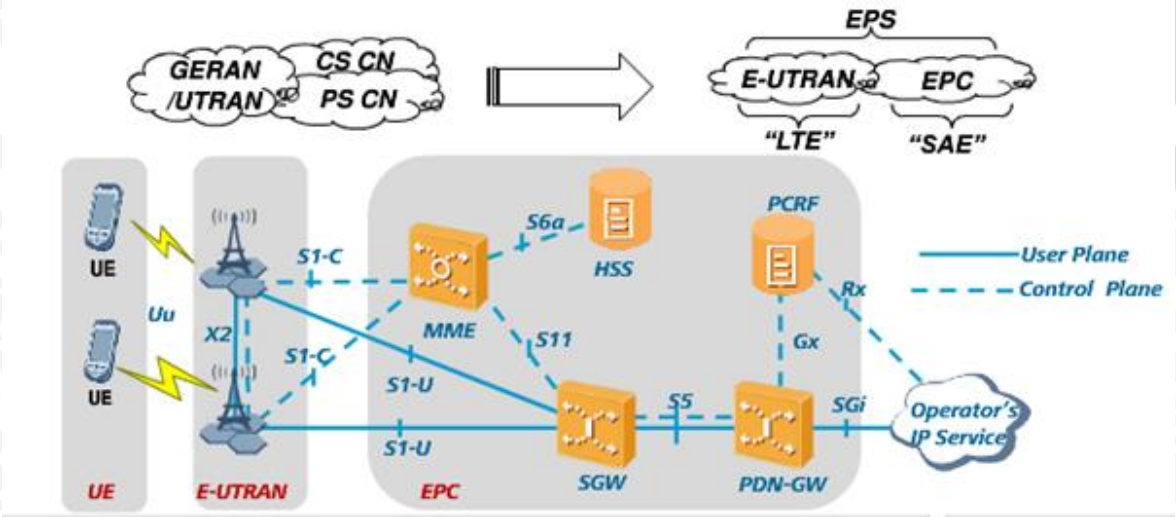
LTE menawarkan beberapa keunggulan dan keuntungan bagi pelanggan dan pihak operator jaringan, yaitu:

1. Efisiensi antenna dan *throughput* yang tinggi, LTE menggunakan OFDM pada arah *downlink,* dimana teknik ini tahan terhadap interferensi akibat lintasan jamak dan menggunakan *single-carrier*- FDMA (SC-FDMA) pada arah *uplink* yang memiliki *peak average power ratio* (PAPR) rendah. Selain itu LTE juga mendukung antenna *multiple input multiple output* (MIMO) yang dapat meningkatkan BER dan *bit rate.*

2. *Latency* yang rendah, jaringan LTE memiliki *setup time* dan *transfer delay* yang sangat rendah, serta waktu *handover* yang rendah .
3. Mendukung bandwidth yang bervariasi, yaitu 1.4, 3, 5, 10, 15 and 20 MHz.
4. Memiliki arsitektur jaringan yang sederhana, hanya ada eNodeB pada *evolved UMTS terrestrial radio access* **(**E-UTRAN).
5. Kompatibel dengan teknologi 3GPP sebelumnya dan teknologi lainnya.
6. Mendukung *frequency division duplex* (FDD) dan *time division duplex* (TDD).

2.3.1 Arsitektur LTE

Arsitektur jaringan LTE dirancang untuk tujuan mendukung trafik *packet switching* dengan mobilitas tinggi, *quality of service* (QOS), dan *latency* yang kecil. Pendekatan *packet switching* ini memperbolehkan semua layanan termasuk layanan *voice* menggunakan koneksi paket. Oleh karena itu pada arsitektur jaringan LTE dirancang sesederhana mungkin, yaitu hanya terdiri dari dua node yaitu eNodeB dan *mobility management entity/gateway* (MME/GW). Hal ini sangat berbeda dengan arsitektur teknologi GSM dan UMTS yang memiliki struktur lebih kompleks dengan adanya *radio network controller* (ENODEB). Beberapa keuntungan yang dapat diperoleh dengan hanya adanya *single node* pada jaringan akses adalah pengurangan *latency* dan distribusi beban proses ENODEB untuk beberapa eNodeB. Pengeliminasian ENODEB pada jaringan akses memungkinkan karena LTE tidak mendukung *soft handover*. Berikut adalah Gambar 2.3 arsitektur LTE.



Gambar 2.3 Arsitektur LTE

(Sumber: An Introduction to LTE, 2009)

Gambar 2.3 merupakan Gambar arsitektur jaringan LTE secara sederhana. Yang dilihat EPC terdiri dari 3 komponen berikut :

1. Serving Gateway (SGW)
2. Packet Data Network Gateway (PDN-GW)
3. Mobility Management Entity (MME)

Bagian E-Utran hanya terdiri dari komponen *Evolved Node B* (eNB). *User Equipment* (UE) merupakan perangkat yang digunakan user untuk berkomunikasi dengan jaringan LTE melalui komponen eNB. UE dapat berupa *handphone, tablet , laptop,* atau perangkat lain yang dilengkapi dengan *network adapter LTE.*

Alur kerja hubungan downlink LTE dimulai dari PD-NGW hingga ke UE. Pada tahap awal, paket data yang berasal dari jaringan di luar jaringan masuk ke jaringan LTE melalu PDN-GW. PDN-GW berfungsi menangani paket-paket data, menetapkan peraturan/izin paket data, penyaringan paket data, pemotongan aliran paket data, dan menghubungkan UE kepada jaringan yang berada di luar jaringan LTE yang biasa disebut dengan IMS (IP *Multimedia Subsystem)* (Poikselk a et al 2012:16). IMS dapat berupa jaringan operator seluler ataupun jaringan internet. PDN-GW juga merupakan pintu masuk dan pintuk keluar bagi setiap paket data yang akan dikirim dari UE, ataupun paket data yang akan diterima UE.

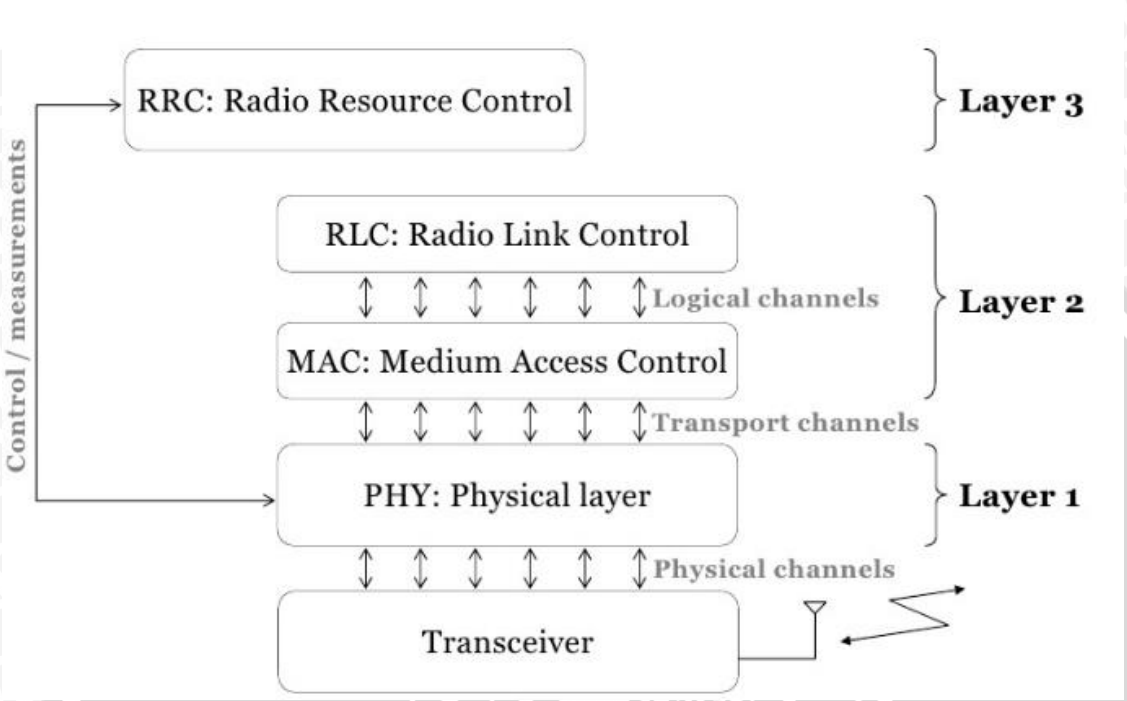
SGW berfungsi sebagai meneruskan paket data Antara eNB dan PDN-GW, setiap UE hanya boleh terhubung kepada satu S-GW. MME adalah komponen yang berfungsi menangani pensinyalan radio seperti mendeteksi status aktivitas UE, melacak keberadaan UE, menangani proses pendaftaran UE sebagai pelanggan dari sebuah operator jaringan seluler, yaitu dengan berkomunikasi dengan HSS, menangani autentikasi UE sebagai pelanggan.

` HSS (*Home Subscriber Server)* adalah yang berfungsi sebagai fungsi pusat penyimpanan informasi data pelanggan dari operator jaringan seluler, seperti informasi langganan data yang dimiliki user dan nomor pelanggan.

eNB adalah komponen yang menangani pengelolaan *radio resource* dan transmisi data langsung kepada UE, proses transmisi data dilakukan dengan menggunakan gelombang radio. Pengguna gelombang radio sebagai media transmisi data menyebabkan dibutuhkannya pengelolaan *radio resource* yang tepat agar semua UE dapat terlayani dengan maksimal (Poikselk a *et al.*, 2012).

2.3.2 Protokol – Protokol LTE

Ketika paket data dikirimkan melalui jaringan LTE, paket data akan melewati node yang mempunyai beberapa protocol atau aturan agar paket data itu dapat terarah ke tempat tujuan dengan benar. Gambar 2.4 adalah *protocol* yang digunakan tiap node pada jaringan LTE.



Gambar 2.4 Protokol LTE
(Sumber: Hyung G. Myung, 2009)

Fungsi protokol dari gambar 2.4 adalah:

a. *Radio Resource Control* (RRC)

   Layanan dan fungsi utama dari RRC adalah *admission control*, manajemen *handover*, manajemen *Quality of Service (*QoS), mengatur dan melaporkan pengukuran stasiun terminal.

b. *Packet Data Convergence Protocol* (PDCP)

   Fungsi utama dari PDCP adalah mengkompress header IP, pengiriman data dari pengguna. Selain itu PDCP bertanggung jawab dalam fungsi pengendalian pengiriman data dan *chipering.* Panjang *header* PDCP 2 byte.

c. *Radio Link Control* (RLC)

   Segmentasi dan *reassembly* paket data dari layer yang lebih tinggi serta perbaikan kesalahan melalu ARQ adalah fungsi utama dari RLC. Selain itu, *flow control* Antara eNodeB dan *mobile terminal* juga ditangani oleh RLC. Panjang Header RLC 2 byte

d. *Medium Access Control* (MAC)

MAC bertanggung jawab untuk penjadwalan *uplink dan downlink*, perbaikan kesalahan melalui hybrid ARQ (HARQ), modulasi adaptif, dan pemetaan antena. Panjang *header* MAC 3 byte

e. Layar fisik (PHY)

Fungsi utama dari layer fisik adalah *coding*, modulasi dan transmisi multiple antena (MIMO)

f. *GPRS Tunnelling Protocol* (GTP)

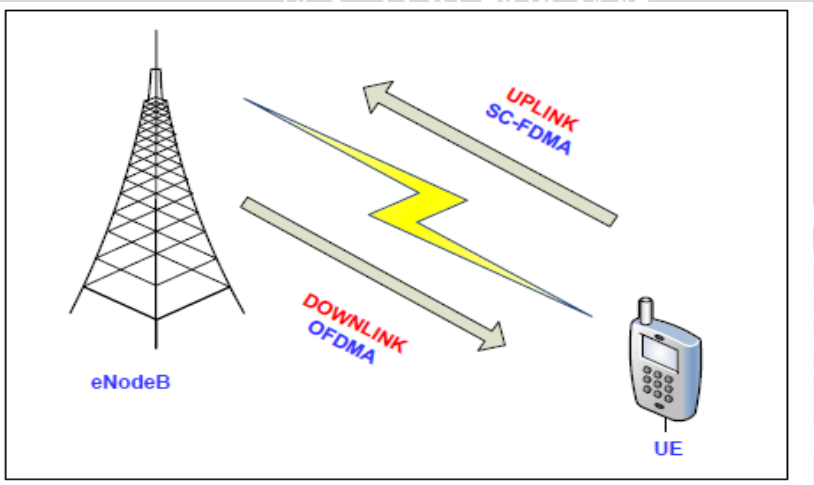
Sebuah protokol *tunneling* khusus disebut GPRS *Tunnelling Protocol* (GTP) yaitu digunakan untuk *interface core network* mempunyai panjang header 8 byte

2.3.3 Interface Radio LTE

Spesifikasi LTE telah ditetapkan oleh 3GPP untuk *user equipment* (UE) dan eNodeB. Adapun spesifikasi teknik LTE yang telah ditetapkan meliputi mode akses radio, teknik akses jamak, mode transmisi MIMO, dan modulasi yang digunakan.

2.3.3.1 Teknik Akses

Pada LTE teknik akses yang digunakan pada transmisi dalam arah *downlink* dan *uplink* berbeda. Arah *downlink* adalah arah komunikasi dari eNodeB ke UE, sementara arah *uplink* adalah arah dari UE menuju eNodeB seperti yang ditunjukkan pada Gambar 2.5. Pada arah *downlink* teknik akses yang digunakan adalah *orthogonal frequency division modulation access* (OFDMA) dan pada arah *uplink* teknik akses yang digunakan adalah *single carrier frequency division multiple access* (SC-FDMA). OFDMA adalah variasi dari *orthogonal frequency division modulation* (OFDM).



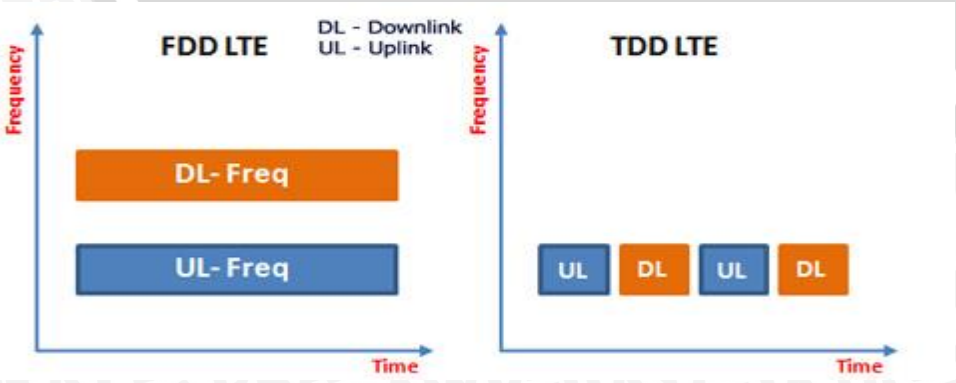
Gambar 2.5 Arah transmisi *downlink* dan *uplink*

(Sumber: Antti Toskala. 2009)

Pada teknik OFDM setiap *subcarrier* adalah *orthogonal* sehingga akan menghemat spektrum frekuensi dan setiap *subcarrier* tidak akan saling mempengaruhi. Akan tetapi salah satu kelemahan teknik akses ini adalah tingginya *peak average power ratio* (PAPR) yang dibutuhkan. Tingginya PAPR dalam OFDM membuat 3GPP melihat skema teknik akses yang berbeda pada arah *uplink* karena akan sangat mempengaruhi konsumsi daya pada UE sehingga pada arah *uplink* LTE menggunakan teknik SC-FDMA. SC-FDMA dipilih karena teknik ini mengkombinasikan keunggulan PAPR yang rendah dengan daya tahan terhadap gangguan lintasan jarak dan alokasi frekuensi yang fleksibel dari OFDMA.

2.3.3.2 Mode Akses Radio

Pada komunikasi seluler sangat penting untuk mempertimbangkan kemampuan jaringan untuk melakukan komunikasi dalam dua arah secara simultan atau dikenal dengan istilah komunikasi *full duplex*. Oleh karena itu untuk dapat melakukan komunikasi dua arah secara simultan, maka dibutuhkan suatu teknik *duplex*. Pada umumnya terdapat dua teknik *duplex* yang biasanya digunakan, yaitu *frequency division duplex* (FDD) dan *time division duplex* (TDD). FDD merupakan teknik *duplex* yang menggunakan dua frekuensi yang berbeda untuk melakukan komunikasi dalam dua arah. Dengan menggunakan FDD dimungkinkan untuk mengirim dan menerima sinyal secara simultan dengan frekuensi yang berbeda-beda. Dengan teknik ini dibutuhkan *guard frequency* untuk memisahkan frekuensi pengiriman dan penerimaan secara simultan, serta dibutuhkan proses *filtering* frekuensi yang harus akurat. Sedangkan TDD menggunakan frekuensi tunggal dan frekuensi tersebut digunakan oleh semua kanal untuk melakukan pengiriman dan penerimaan data. Setiap kanal tersebut di-*multiplexing* dengan menggunakan basis waktu sehingga setiap kanal memiliki *time slot* yang berbeda. Perbedaan teknik FDD dan TDD dapat dilihat pada Gambar 2.6



Gambar 2.6 FDD dan TDD pada LTE

(Sumber: An Introduction to LTE, 2009)

Pada Gambar 2.6 dapat dilihat bahwa dalam teknik FDD lebih banyak menggunakan spektrum frekuensi yang tersedia. FDD lebih unggul dalam menangani *latency* dibandingkan TDD karena kanal harus lebih lama menunggu waktu pemprosesan dalam *multiplexing*.

Interface radio LTE mendukung *frequency divison duplex* dan *time divison duplex* (TDD), yang masing-masing memiliki struktur *frame* yang berbeda-beda. Pada LTE terdapat 15 band operasi FDD dan 8 band operasi TDD pada LTE. LTE juga dapat menggunakan fasilitas *half-duplex* FDD yang mengizinkan *sharing* hardware di antara *uplink* dan *downlink* dimana koneksi *uplink* dan *downlink* tidak digunakan secara simultan. LTE dapat menggunakan kembali semua *band* frekuensi yang digunakan pada UMTS. Berikut tabel 2.4 tentang pembagian band frekuensi LTE mode FDD dan TDD

Tabel 2.4 Spektrum Frekuensi LTE TDD dan FDD

E-UTRA Operating Band	Uplink (UL) operating band BS receive UE transmit	Downlink (DL) operating band BS transmit UE receive	Duplex Mode
	F_{UL_low} – F_{UL_high}	F_{DL_low} – F_{DL_high}	
1	1920 MHz – 1980 MHz	2110 MHz – 2170 MHz	FDD
2	1850 MHz – 1910 MHz	1930 MHz – 1990 MHz	FDD
3	1710 MHz – 1785 MHz	1805 MHz – 1880 MHz	FDD
4	1710 MHz – 1755 MHz	2110 MHz – 2155 MHz	FDD
5	824 MHz – 849 MHz	869 MHz – 894MHz	FDD
6	830 MHz – 840 MHz	875 MHz – 885 MHz	FDD
7	2500 MHz – 2570 MHz	2620 MHz – 2690 MHz	FDD
8	880 MHz – 915 MHz	925 MHz – 960 MHz	FDD
9	1749.9 MHz – 1784.9 MHz	1844.9 MHz – 1879.9 MHz	FDD
10	1710 MHz – 1770 MHz	2110 MHz – 2170 MHz	FDD
11	1427.9 MHz – 1447.9 MHz	1475.9 MHz – 1495.9 MHz	FDD
12	698 MHz – 716 MHz	728 MHz – 746 MHz	FDD
13	777 MHz – 787 MHz	746 MHz – 756 MHz	FDD
14	788 MHz – 798 MHz	758 MHz – 768 MHz	FDD
17	704 MHz – 716 MHz	734 MHz – 746 MHz	FDD
...			
33	1900 MHz – 1920 MHz	1900 MHz – 1920 MHz	TDD
34	2010 MHz – 2025 MHz	2010 MHz – 2025 MHz	TDD
35	1850 MHz – 1910 MHz	1850 MHz – 1910 MHz	TDD
36	1930 MHz – 1990 MHz	1930 MHz – 1990 MHz	TDD
37	1910 MHz – 1930 MHz	1910 MHz – 1930 MHz	TDD
38	2570 MHz – 2620 MHz	2570 MHz – 2620 MHz	TDD
39	1880 MHz – 1920 MHz	1880 MHz – 1920 MHz	TDD
40	2300 MHz – 2400 MHz	2300 MHz – 2400 MHz	TDD

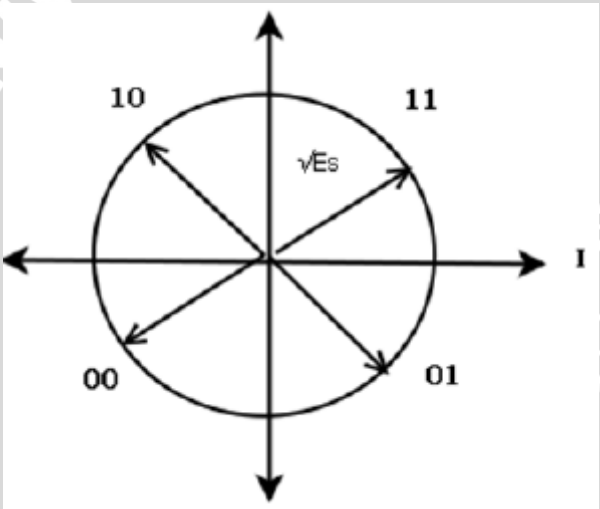
(Sumber: Bong Youl, 2012)

2.3.4 Modulasi LTE (*Long Term Evolution*)

Ada beberapa tipe modulasi yang digunakan pada jaringan *Long Term Evolution (LTE)* adalah:

1. Modulasi QPSK

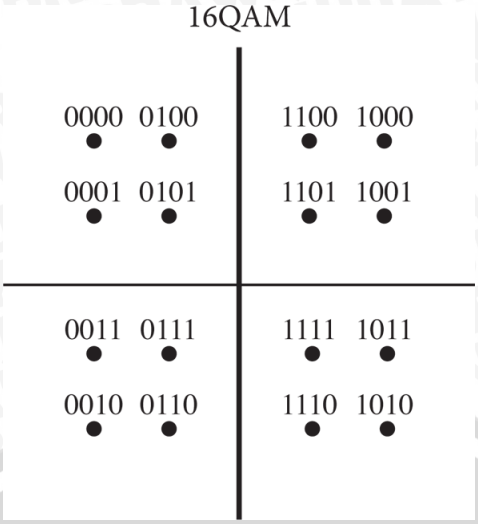
QPSK dikenal juga dengan *quadrature phase shift keying* atau 4 PSK. QPSK menggunakan 4 titik dengan *konstelasi* dengan mengirimkan 2 bit untuk setiap symbol (00, 01, 10, 11). Dengan menggunakan 4 fasa, QPSK dapat melakukan encode 2 bit per *symbol.* Dalam QPSK ada empat fasa keluaran yang berbeda, maka harus ada empat kondisi masukan yang berbeda pula. Keunggulan QPSK adalah mampu mentransmisikan data 2 kali lebih cepat dibandingkan BPSK dan lebih efisien dalam penggunaan spectrum frekuesni. Sedangkan kelemahannya adalah kurang tahan terhadap noise dan rawan terjadi interferensi dari sinyal lain.



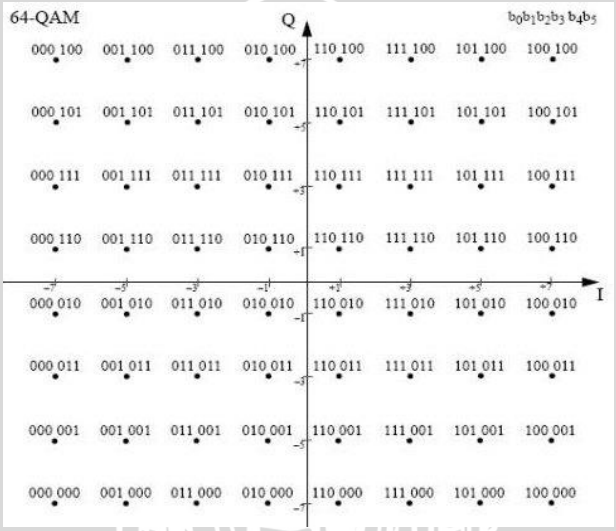
Gambar 2.7 Diagram Konstelasi pada QPSK
(Sumber: Fuqin Xiong, 2006)

2. Modulasi QAM

QAM merupakan tekmik modulasi yang merupakan perpaduan Antara ASK dan PSK. Sehingga pada QAM, amplitude dan fasa sinyal *carrier* akan berhubung terhadap perubahan amplitudo sinyal informasi, yang mengakibatkan sinyal direpresentasikan dalam besaran amplitudo dan pergeseran fasa. Modulasi QAM membawa data dengan merubah parameter dari sinyal *carrier* untuk merespon sinyal. Dalam hal ini, amplitude 2 sinyal *carrier* QAM yang berbeda 90 derajat Antara satu dengan yang lainnya dirubah untuk mendapatkan sinyal yang diinginkan. Jenis Modulasi QAM yang digunakan dalam teknologi LTE adalah 16-QAM dan 64- QAM.



Gambar 2.8 Diagram Konstelasi 16-QAM
(Sumber: Fuqin Xiong, 2006)



Gambar 2.9 Diagram Konstelasi 64-QAM
(Sumber: Fuqin Xiong, 2006)

2.4 Parameter Kinerja Jaringan

Quality of Service (QoS) adalah parameter yang menyatakan kinerja suatu jaringan berdasarkan beberapa parameter. Sedangkan menurut ITU-T E.800, QoS adalah sekumpulan efek performansi yang menentukan derajat kepuasan pengguna terhadap *service* yang diberikan oleh jaringan. Pada jaringan LTE, QoS dapat dilihat dari parameter *delay*, *throughput* dan *packet loss* yang dilihat dari sisi pengguna.

2.4.1 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 2.5.

Tabel 2.5 Spesifikasi Audio dan Video *Codec*

Video *CODEC*	***Bit Rate* (kbps)**	***Maximum Payload* (byte)**	***Delay CODEC* (ms)**
AVC/H.264 MPEG4	96-384	254	16-50
Audio *CODEC*	***Bit Rate* (kbps)**	***Maximum Payload* (byte)**	***Delay CODEC* (ms)**
AAC	128	63	24-60

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

Besarnya nilai *delay codec* pada aplikasi *video on demand* penelitian ditunjukkan persamaan 2.5. (Sirtufillaila, 2010)

$$t_{codec} = t_{audio} + t_{video} \quad (2\text{-}1)$$

Keterangan:

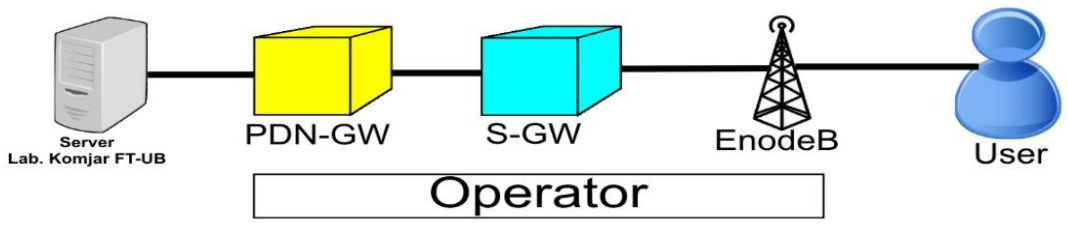
t_{codec} = *Delay codec* keseluruhan (s)

t_{audio} = *Delay codec* audio (s)

t_{video} = *Delay codec* video (s)

2.4.2 *Delay End-to-End* pada LTE

Delay adalah waktu yang dibutuhkan untuk mengirimkan paket data dari sumber sampai ke tujuan. *Delay* pada jaringan LTE 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*.



Gambar 2.10 *Delay* pada LTE

(Sumber: Hasil Perancangan)

Delay End-to-End berdasarkan gambar dihitung dengan menggunakan persamaan:

$$t_{end\ to\ end} = t_{proses} + t_t + t_p + t_w \quad (2\text{-}2)$$

Keterangan:

$t_{end\ to\ end}$ = *delay end-to-end* pada jaringan LTE (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)
$t_{Paketisasi}$ = *delay* paketisasi (s)
$t_{depaketisasi}$ = *delay* depaketisasi (s)

2.4.2.1 *Delay* Proses

Delay proses adalah *delay* yang terjadi saat paket dienkapsulasi dan didekapsulasi pada suatu node. Enkapsulasi adalah proses pemaketan data dengan menambahkan sebuah *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 untuk proses dekapsulasi atau membaca keseluruhan *header* dari sebuah paket yang diterima suatu node.

Besarnya *delay* dekapsulasi dan *delay* enkapsulasi dapat diperoleh dengan menggunakan persamaan di bawah ini [Onno W. Purbo, et al., 2001: 24]:

$$t_{enc} = \frac{W_{frame} - L}{C} \times 8 \quad (2\text{-}3)$$

$$t_{dec} = \frac{W_{frame} - L}{C} \times 8 \quad (2\text{-}4)$$

Keterangan:

t_{enc} = *delay* enkapsulasi (s)
t_{dec} = *delay* dekapsulasi (s)
W_{frame} = panjang *frame* (byte)
L = panjang paket data di *node* (byte)
C = kecepatan pemrosesan data (bps)

Delay enkapsulasi pada LTE terjadi pada titik server, PDN-GW, S-GW dan eNode B. Sedangkan *delay* dekapsulasi pada LTE terjadi pada titik PDN-GW, S-GW, eNode B dan UE. Adapun besarnya *delay* enkapsulasi pada LTE didapatkan dengan menggunakan persamaan sebagai berikut,

$$t_{enc} = t_{e-server} + t_{e-PDN-GW} + t_{e-S-GW} + t_{e-ENODEB} \quad (2\text{-}5)$$

Keterangan:

t_{enc} = delay enkapsulasi total (s)

$t_{e\text{-}server}$ = delay enkapsulasi server (s)

$t_{e\text{-}PDN\text{-}GW}$ = delay enkapsulasi PDN-GW (s)

$t_{e\text{-}S\text{-}GW}$ = delay enkapsulasi S-GW (s)

$t_{e\text{-}NodeB}$ = delay enkapsulasi eNode-B (s)

Kemudian besarnya *delay* dekapsulasi pada LTE didapatkan dengan menggunakan persamaan sebagai berikut,

$$t_{dec} = t_{d-PDN-GW} + t_{d-S-GW} + t_{d-ENODEB} + t_{d-UE} \quad (2\text{-}6)$$

Keterangan:

t_{dec} = delay dekapsulasi total (s)

$t_{d\text{-}PDN\text{-}GW}$ = delay dekapsulasi PDN-GW (s)

$t_{d\text{-}S\text{-}GW}$ = delay dekapsulasi S-GW (s)

$t_{d\text{-}ENODEB}$ = delay dekapsulasi eNode-B (s)

$t_{d\text{-}UE}$ = delay dekapsulasi UE (s)

Sehingga besarnya *delay* proses pada jaringan LTE adalah sebagai berikut.

$$t_{proses} = t_{enc} + t_{dec} \quad (2\text{-}7)$$

- **Server**

Untuk mempermudah perhitungan secara teoretis, Jaringan IP *network* dianggap sebagai server yang terletak di antara *media server* penyedia layanan dan perangkat pengguna pada penelitian ini.

Dalam perhitungan diasumsikan bahwa besarnya data ketika data tersebut telah melewati *layer* aplikasi dan menuju ke *transport layer*, sehingga data akan menjadi paket yang disebut segmen. Pada penelitian ini protokol aplikasi yang digunakan adalah RTP/RTCP untuk men*download* file *streaming video* yang bersifat *real time*, maka digunakan *header* UDP. Paket data *video on demand* ditambahkan *header* RTP, UDP, NALU dan IP pada layer *Transport*.

Untuk menghitung besar *payload* video dan *payload* audio *file video on demand* ditentukan dengan persamaan 2.12 dan 2.13. (Sirtufillaila, 2010)

$$Pv = bitrate \text{ video x } frame\ size \text{ video} \quad (2\text{-}8)$$

$$\text{Pa} = bitrate \text{ audio x } frame\ size \text{ audio} \quad (2\text{-}9)$$

Keterangan:

Pv = Besarnya paket data video pada *file* (bit)

Pa = Besarnya paket data audio pada *file* (bit)

Bitrate video = Besar kecepatan pengolahan bit video (bps)

Bitrate audio = Besar kecepatan pengolahan bit video (bps)

Frame Size video = Lama waktu 1 *frame* video (s)

Frame Size audio = Lama waktu audio untuk 1 *frame* 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.

$$\text{Nv} = \frac{\text{Pv}}{\text{Pv maksimumx 8}} \quad (2\text{-}10)$$

$$\text{Na} = \frac{\text{Pa}}{\text{Pa maksimum x 8}} \quad (2\text{-}11)$$

Keterangan:

Nv = Jumlah paket video

Na = Jumlah paket audio

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

Pa 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 dienkodekan dan ditambah *header* sesuai dengan persamaan 2.16 dan 2.17.

$$\text{Pv } (encoded) = \text{Pv} + \text{Nv x } (\text{H}_{\text{NALU}} + \text{H}_{\text{RTP}} + \text{H}_{\text{UDP}} + \text{H}_{\text{IPv4}}) \quad (2\text{-}12)$$

Dan untuk audio,

$$\text{Pa } (encoded) = \text{Pa} + \text{Na x } (\text{H}_{\text{NALU}} + \text{H}_{\text{RTP}} + \text{H}_{\text{UDP}} + \text{H}_{\text{IPv4}}) \quad (2\text{-}13)$$

Keterangan:

Pv (*encoded*) = *Payload* video setelah penambahan *header* NALU, RTP, UDP, IP (byte)

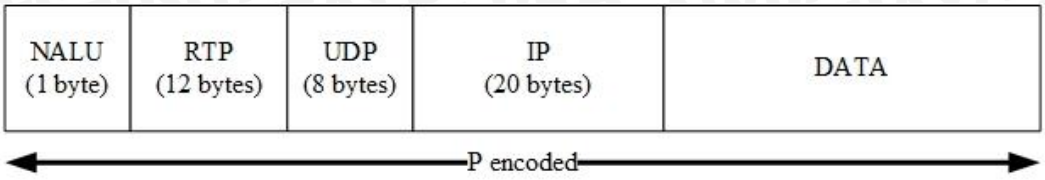
Pa (*encoded*) = *Payload* audio setelah penambahan *header* NALU, RTP, UDP, IP (byte)

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*
Sumber: Analisis

Besarnya paket data aplikasi *video on demand* yang ditransmisikan pada sistem ditentukan dengan persamaan 2.18.

$$P_{vod\ size} = Pv(encoded) + Pa(encoded) \quad (2\text{-}14)$$

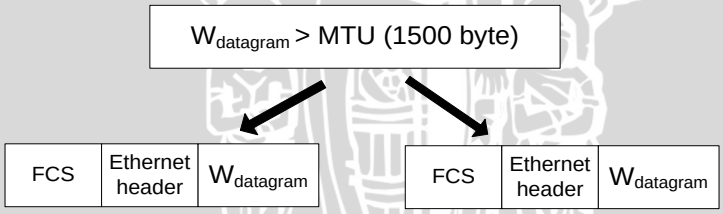
Keterangan:

P*VoD size* = Panjang datagram VoD (byte)

Pv (*encoded*) = *Payload* video setelah penambahan *header* NALU, RTP, UDP, IP (byte)

Pa (*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 akan langsung ditambahkan dengan *header Ethernet* dan FCS (Drew Heywood dan Rob Scrimger, 1997).



Gambar 2.12 Format Segmentasi Datagram IP
Sumber: Taufan Riza, 2002:41

Jika datagram IP melebihi *Maximum Transmission Unit* (MTU) *Ethernet* sebesar 1.500 byte, maka datagram IP tersebut akan disegmentasi terlebih dahulu baru ditambahkan dengan *header Ethernet* sesuai dengan persamaan sebagai berikut,

$$N_{\text{frame Ethernet}} = \frac{P_{VoD-size}}{MTU_{ethernet}} \quad (2\text{-}15)$$

Keterangan:

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

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

$P_{VoD\text{-}size}$ = Besar paket *video on demand* (byte)

Maka, jumlah *frame* yang dikirimkan server menuju PDN-GW adalah sebagai berikut,

$$W_{\text{frame server}} = P_{\text{VoD}-\text{size}} + [N_{\text{frame Ethernet}} \text{ x } (H_{\text{Ethernet}} + \text{FCS})] \quad (2\text{-}16)$$

Keterangan:

$W_{\text{frame server}}$ = Datagram *frame server* (byte)

H_{Ethernet} = Panjang *header* Ethernet (14 byte)

FCS = Panjang *Frame Check Sequence* (4 byte)

Sehingga *delay* enkapsulasi pada server dapat ditentukan dengan persamaan sebagai berikut,

$$t_{e-server} = \frac{W_{\text{frame server}} - P_{\text{VoD}-\text{size}}}{C_{\text{server}}} \text{x8} \quad (2\text{-}17)$$

Keterangan:

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

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

Setelah dari server paket data akan menuju ke PDN-GW. Proses yang dilakukan paket pada di PDN-GW yang pertama adalah melakukan dekapsulasi data, yaitu melepas header dari server. Sehingga panjang data VoD pada PDN-GW sesuai persamaan 2.17,

$$W_{\text{VOD PDN}-\text{GW}} = W_{\text{frame server}} - H_{\text{Ethernet}} - \text{FCS} \quad (2\text{-}18)$$

Keterangan:

$W_{\text{VoD PDN-GW}}$ = Paket data *video on demand* di PDN-GW (byte)

H_{UDP} = *Header* UDP (8 byte)

H_{IPv4} = *Header* IP (20 byte)

H_{Ethernet} = *Header* Ethernet (14 byte)

FCS = *Header* FCS (4 byte)

Kemudian besarnya *delay* dekapsulasi yang terjadi di titik PDN-GW ditunjukkan oleh persamaan sebagai berikut,

$$t_{d-\text{PDN}-\text{GW}} = \frac{W_{\text{frame sever}} - W_{\text{VOD PDN}-\text{GW}}}{C_{\text{PDN}-\text{GW}}} \text{x8} \quad (2\text{-}19)$$

Keterangan:

$t_{\text{d- PDN-GW}}$ = Delay dekapsulasi pada PDN-GW (s)

$W_{\text{VoD PDN-GW}}$ = Panjang data pada PDN-GW (byte)

$C_{\text{PDN-GW}}$ = Kecepatan transmisi data di PDN-GW (bps)

Paket data yang telah didekapsulasi dan diterima oleh PDN-GW, selanjutnya dienkapsulasi kembali untuk dikirim menuju node S-GW. Pada PDN-GW terdapat ukuran maksimum *frame,* yakni MSS. MSS adalah *Maximum Segment Size*. Nilai MSS didapatkan dari,

$$\mathrm{MSS} = \mathrm{MTU} - \mathrm{H_{GTP}} - \mathrm{H_{UDP}} - \mathrm{H_{IPv4}} \quad (2\text{-}20)$$

Keterangan:

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

Datagram *video on demand* pada PDN-GW kemudian disegmentasi dengan MSS, jika datagram tersebut melebihi ukuran MSS. Paket data yang disegmentasi oleh MSS ditentukan dengan persamaan sebagai berikut,

$$\mathrm{N_{datagram}} = \frac{\mathrm{W_{VoD\ PDN-GW}}}{\mathrm{MSS}} \quad (2\text{-}21)$$

Keterangan:

$\mathrm{N_{datagram}}$ = Jumlah *frame* pada PDN-GW

Kemudian datagram *video on demand* dienkapsulasi kembali dengan penambahan header GTP, UDP dan IP sesuai dengan persamaan sebagai berikut,

$$\mathrm{W_{datagram\ PDN-GW}} = \mathrm{W_{VoD\ PDN-GW}} + \mathrm{N_{datagram}}\ \mathrm{x}\ (\mathrm{H_{GTP}} + \mathrm{H_{UDP}} + \mathrm{H_{IPv4}} + \mathrm{H_{Ethernet}} + \mathrm{FCS}) \quad (2\text{-}22)$$

Keterangan:

$\mathrm{W_{datagram\ PDN-GW}}$ = Panjang total datagram PDN-GW (byte)

$\mathrm{W_{VoD\ PDN-GW}}$ = Panjang data pada PDN-GW (byte)

$\mathrm{H_{GTP}}$ = Panjang *header* GTP (8 byte)

$\mathrm{H_{UDP}}$ = Panjang *header* UDP (8 byte)

$\mathrm{H_{IPv4}}$ = Panjang *header* IP (20 byte)

Untuk dikirimkan menuju S-GW, maka datagram PDN-GW disegmentasi kembali jika melebihi MTU (1500 byte). Persamaan segmentasi datagram PDN-GW dengan MTU *Ethernet* ditunjukkan oleh persamaan sebagai berikut,

$$\mathrm{N_{frame\ Ethernet\ PDN-GW}} = \frac{\mathrm{W_{datagram\ PDN-GW}}}{\mathrm{MTU_{Ethernet}}} \quad (2\text{-}23)$$

Keterangan:

$\mathrm{N_{frame\ Ethernet\ PDN-GW}}$ = Jumlah frame PDN-GW

Sehingga jumlah keseluruhan *frame* PDN-GW yang dikirim menuju S-GW dapat diketahui dengan persamaan sebagai berikut,

$$W_{\text{frame PDN-GW}} = W_{\text{datagram PDN-GW}} + [N_{\text{frame Ethernet PDN-GW}} \, x(H_{\text{Ethernet}} + FCS)] \quad (2\text{-}24)$$

Keterangan:

$W_{\text{frame PDN-GW}}$ = Panjang frame PDN-GW (byte)

Sehingga *delay* enkapsulasi yang terjadi pada PDN-GW ditunjukkan dengan persamaan sebagai berikut,

$$t_{e-PDN-GW} = \frac{W_{\text{frame PDN-GW}} - W_{\text{VoD PDN-GW}}}{C_{PDN-GW}} \, x8 \quad (2\text{-}25)$$

- ***Service Gateway*** (S-GW)

Paket data yang diterima dari PDN-GW didekapsulasi kemudian dienkapsulasi untuk dikirimkan menuju *node* selanjutnya. Paket data yang diterima pada S-GW dari PDN-GW ditunjukkan dengan persamaan sebagai berikut,

$$W_{\text{VoD S-GW}} = W_{\text{frame-PDN-GW}} - H_{GTP} - H_{UDP} - H_{IPv4} - H_{\text{Ethernet}} - FCS \quad (2\text{-}26)$$

Keterangan:

$W_{\text{VoD S-GW}}$ = Panjang paket data *video on demand* di PDN-GW (byte)

Kemudian besarnya *delay* dekapsulasi pada S-GW didapatkan dengan persamaan sebagai berikut,

$$t_{dS-GW} = \frac{W_{\text{frame S-GW}} - W_{\text{VoD S-GW}}}{C_{S-GW}} x8 \quad (2\text{-}27)$$

Keterangan:

$C_{\text{S-GW}}$ = Kecepatan transmisi data pada S-GW (bps)

Apabila Datagram *video on demand* pada S-GW melebihi *Maximum Transfer Unit* (MTU) *Ethernet* sebesar 1500 byte, maka paket data *video on demand* akan disegmentasi

kembali. Persamaan segmentasi datagram S-GW dengan MTU *Ethernet* ditunjukkan oleh persamaan sebagai berikut,

$$N_{Frame\ S-GW} = \frac{W_{VoD\ S-GW}}{MTU} \quad (2\text{-}28)$$

Keterangan:

$N_{frame\ S-GW}$ = Jumlah *frame* pada S-GW

Kemudian datagram *video on demand* dienkapsulasi kembali dengan penambahan header GTP, UDP dan IP sesuai dengan persamaan sebagai berikut,

$$W_{datagram\ S-GW} = W_{VoD\ S-GW} + N_{Frame\ S-GW}\ x\ (H_{GTP} + H_{UDP} + H_{IPv4} + H_{Ethernet} + FCS) \quad (2\text{-}29)$$

Keterangan:

$W_{datagram\ S-GW}$ = Panjang datagram S-GW (byte)

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

$$t_{e\ S-GW} = \frac{W_{datagram\ S-GW} - W_{VoD\ S-GW}}{C_{S-GW}}\ x\ 8 \quad (2\text{-}30)$$

- ***Evolved Node B*** **(ENodeB)**

Sama seperti proses pada PDN-GW dan S-GW, paket data diterima pada eNodeB kemudian dilakukan proses dekapsulasi dan enkapsulasi. Proses dekapsulasi dilakukan terlebih dahulu saat paket data dari S-GW diterima oleh ENODEB. Paket data yang diterima ENODEB dari S-GW ditunjukkan dengan persamaan,

$$W_{VoD\ ENODEB} = W_{datagram\ S-GW} - H_{GTP} - H_{UDP} - H_{IPv4} - H_{Ethernet} - FCS \quad (2\text{-}31)$$

Keterangan:

$W_{VoD\ ENODEB}$ = panjang paket data video on demand di ENODEB (byte)

Sehingga nilai *delay* dekapsulasi pada ENODEB dapat ditentukan dengan persamaan sebagai berikut,

$$t_{d-ENODEB} = \frac{W_{datagram\ S-GW} - W_{VoD\ ENODEB}}{C_{ENODEB}} x8 \quad (2\text{-}32)$$

Keterangan:

C_{ENODEB} = Kecepatan transmisi data pada ENODEB (Bps)

Packet Data Convergence Protocol (PDCP) adalah *layer* ENODEB yang berfungsi mengenkapsulasi datagram IP *video on demand* pada ENODEB. Adapun persamaan untuk enkapsulasi *video on demand* adalah sebagai berikut,

$$N_{\text{datagram ENODEB}} = \frac{W_{\text{VoD ENODEB}}}{1500} \quad (2\text{-}33)$$

Keterangan:

$N_{\text{datagram ENODEB}}$ = Jumlah *frame* ENODEB

Kemudian besar *frame* total pada *layer* PDCP dapat ditentukan dengan persamaan sebagai berikut,

$$W_{\text{frame PDCP}} = W_{\text{VoD ENODEB}} + H_{\text{PDCP}} \quad (2\text{-}34)$$

Keterangan:

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

H_{PDCP} = panjang *header* PDCP (2 byte)

Selanjutnya pada *layer Radio Link Control* (RLC), *frame* PDCP disegmentasi menjadi RLC PDU *fixed size* sebesar 40 byte [Anthony Lo, 2004:8]. Sehingga persamaan untuk menentukan jumlah *frame* RLC adalah sebagai berikut,

$$N_{\text{frame RLC}} = \frac{W_{\text{frame PDCP}}}{40 \text{ byte}} \quad (2\text{-}35)$$

Keterangan:

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

Panjang setiap *frame* RLC adalah sebanyak 2 byte karena setiap *frame* selanjutnya diberi *header* sebanyak 2 byte. Adapun panjang *frame* RLC total yang diteruskan menuju *layer Medium Access Control* (MAC) ditentukan dengan persamaan sebagai berikut,

$$W_{\text{frame RLC total}} = W_{\text{frame PDCP}} \text{ x } H_{\text{frame RLC}} \quad (2\text{-}36)$$

Keterangan:

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

$H_{\text{frame RLC}}$ = panjang *frame* RLC (2 byte)

Pada saat sampai di *layer* MAC-d, RLC PDU disegmentasi menjadi MAC-d *Service Data Unit* (MAC SDU) sebesar 42 byte. Jumlah *frame* MAC SDU ditentukan dengan persamaan sebagai berikut,

$$N_{frame\ MAC} = \frac{W_{frame\ RLC\ total}}{42\ byte} \quad (2\text{-}37)$$

Keterangan:

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

Sedangkan panjang *frame* MAC-d ditentukan dengan persamaan sebagai berikut,

$$W_{frame\ MAC} = W_{frame\ RLC\ total} + (N_{frame\ MAC}\ x\ H\ mac) \quad (2\text{-}38)$$

Keterangan:

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

$H_{MAC\text{-}d}$ = panjang *header* MAC (3 byte)

Sehingga besar *delay* dekapsulasi yang terjadi pada ENODEB dapat ditentukan dengan persamaan sebagai berikut,

$$t_{e-ENODEB} = \frac{W_{frame\ MAC} - W_{VoD\ ENODEB}}{C_{ENODEB}} x8 \quad (2\text{-}39)$$

Keterangan:

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

- ***User Equipment* (UE)**

Pada *user equipment* terjadi proses dekapsulasi. Proses dekapsulasi memproses paket data dari Node B yaitu melepas *header – header* dari EnodeB dan dimodelkan dengan persamaan sebagai berikut,

$$W_{VoD\ UE} = W_{frame\ MAC} - (N_{frame} x H_{PDCP}) - (N_{frame} x H_{RLC}) - (N_{frame} x H_{MAC}) \quad (2\text{-}40)$$

Keterangan:

$W_{VoD\ UE}$ = Panjang paket data *video on demand* pada UE (byte)

Maka, besarnya *delay* dekapsulasi pada UE dapat ditentukan sebagai berikut,

$$t_{d-UE} = \frac{W_{frame\ MAC} - W_{VoD\ UE}}{C_{UE}} x8 \quad (2\text{-}41)$$

Keterangan:

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

2.4.2.2 *Delay* Transmisi

Delay transmisi adalah waktu yang diperlukan sebuah paket data untuk melalui media transmisi dari satu node ke node yang lain. Pada delay ini akan dipengaruhi oleh panjang paket data yang dikirim dan kecepatan jenis media transmisi yang digunakan. Pada skripsi ini VoD yang akan dikirim terdiri dari paket audio dan video. *Delay* transmisi terjadi ketika paket data akan diletekan pada media transmisi dan akan berjalan menuju titik lainnya, maka delay ini terjadi Antara server ke PDN-GW, PDN-GW ke S-GW, S-GW ke EnodeB, dan EnodeB ke user

Berdasarkan Mischa Swartz, delay transmisi dapat ditentukan dengan persamaan sebagai berikut (Mischa Schwartz, 1987: 132),

$$t_T = \frac{W}{C} \quad (2\text{-}42)$$

Keterangan:

t_T = delay transmisi (s)

W = panjang *frame* pada suatu *node* (byte)

C = kecepatan transmisi (bps)

Sehingga delay transmisi total pada LTE, dapat ditentukan dengan persamaan sebagai berikut,

$$t_{T\,total} = t_{T1} + t_{T2} + t_{T3} + t_{T4} \quad (2\text{-}43)$$

Keterangan:

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

t_{T1} = *delay* transmisi antara Server - PDN-GW (s)

t_{T2} = *delay* transmisi antara PDN-GW – S-GW (s)

t_{T3} = *delay* transmisi antara S-GW - ENODEB(s)

t_{T4} = *delay* transmisi antara ENODEB-UE(s)

2.4.2.3 *Delay* Propagasi

Menurut Forouzan dalam bukunya *Data Communications and Networking* tahun 2000, *delay* propagasi adalah waktu yang dibutuhkan oleh paket data untuk merambat dari node S-GW menuju UE. Persamaannya ditentukan sebagai berikut,

$$t_p = \frac{d_{max}}{v} \quad (2\text{-}44)$$

Keterangan:

t_p = *delay* propagasi (s)

d_{max} = jarak atau radius sel (m)

V = cepat rambang gelombang elektromagnetik (3 x 10^8 m/s)

2.4.2.4 *Delay* Antrian

Delay antrian adalah waktu di mana paket data berada dalam antrian untuk ditransmisikan. *Delay* antrian dapat dihitung dengan menggunakan model antrian M/M/1. Delay antrian adalah waktu sebuah paket data menunggu sampai paket lain ditransmisikan. M pertama menunjukkan kedatangan *Poisson*, M kedua berarti distribusi waktu pelayanan eksponensial, dan 1 menunjukkan bahwa jumlah node yang melayani adalah tunggal.

Delay antrian pada jaringan LTE terjadi pada titik server, PDN-GW, S-GW, eNodeB. Besarnya *delay* antrian tersebut ditentukan oleh persamaan sebagai berikut, (Mischa Schwartz, 1987)

$$t_w = t_{queue} + t_{serv} \quad (2\text{-}45)$$

sedangkan,

$$t_{queue} = \frac{\lambda}{\mu(\mu-\lambda)} \quad (2\text{-}46)$$

dan,

$$t_{serv} = \frac{1}{\mu} \quad (2\text{-}47)$$

dan,

$$\mu = \frac{C}{L} \quad (2\text{-}48)$$

dan,

$$\lambda = \rho \, . \, \mu \quad (2\text{-}49)$$

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 dengan kenaikan sebesar 0,1. Hal ini dikarenakan jika $\rho > 1$ berarti rata-rata kedatangan melampaui rata-rata pelayanan, sehingga panjang antrian bertambah tanpa batas.

Sehingga *delay* antrian total pada LTE dapat ditentukan dengan persamaan sebagai berikut,

$$t_{w\,total} = t_{w\,server} + t_{w\,S-GW} + t_{w\,ENODEB} + t_{w\,PDN-GW} \quad (2\text{-}50)$$

Keterangan:

$t_{w\,total}$ = *delay* antrian total (s)
$t_{w\,PDN-GW}$ = *delay* antrian pada PDN-GW (s)
$t_{w\,S-GW}$ = *delay* antrian pada S-GW (s)
$t_{w\,ENODEB}$ = *delay* antrian pada ENODEB (s)
$t_{w\,Server}$ = *delay* antrian pada Server (s)

2.4.2.5 *Delay* Paketisasi dan Depaketisasi

Delay paketisasi adalah waktu yang dibutuhkan untuk memaketkan data *audio* dan video. Delay ini hanya terjadi sekali di *source* informasi. Sedangkan *delay* depaketisasi adalah waktu yang dibutuhkan penerima untuk mengubah paket-paket yang diterima menjadi data yang bisa dibaca. *Delay* ini dipengaruhi paket data dan kecepatan pemrosesan data. *Delay* paketisasi dinyatakan dengan persamaan 2.46

$$t_{Paketisasi} = \frac{L\,paket}{C} \quad (2\text{-}51)$$

$L_{Paketisasi}$ = besar paket yang dikirimkan (byte)
C = kapasitas kanal transmisi (bps)

Sedangkan untuk *delay* depaketisasi dinyatakan dengan persamaan 2.47

$$t_{Depaketisasi} = \frac{L\,paket}{C} \quad (2\text{-}52)$$

$L_{Depaketisasi}$ = besar paket yang diterima (byte)
C = Kapasitas kanal transmisi (bps)

2.4.3 Probabilitas *Packet Loss* pada LTE

Probabilitas *packet loss* merupakan besarnya kemungkinan jumlah paket data yang hilang pada saat transmisi. Probabilitas *packet loss* total pada LTE ditentukan berdasarkan

probabilitas *packet loss* pada jaringan tersebut serta pada *transport layer* UDP dan *network layer* seperti pada persamaan 2.73. (Anurag, 2008)

$$\rho_{tot} = 1 - [(1 - \rho_{net})(1 - \rho_{vod})] \quad (2\text{-}53)$$

dengan:

ρ_{tot} = Probabilitas *packet loss* total

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

$\rho_{TCP/IP}$ = Probabilitas *packet loss* p_{vod}

Besarnya probabilitas *packet loss* dalam persen dirumuskan dalam persamaan sebagai berikut (Wijaya Hendra A, 2005)

$$\text{packet loss } (\%) = \frac{N_{\text{packet loss}}}{N_{packet\ loss} + N_{Paket}} \text{ x } 100\% \quad (2\text{-}54)$$

dengan :

$N_{packet\ loss}$ = jumlah paket yang hilang

N_{paket} = jumlah paket yang diterima dengan benar

a) Probabilitas *Packet Loss* pada VoD

Packet Loss yang terjadi pada layanan *VoD* dihitung melalui persamaan

$$\rho = P_{size.}\ \rho_b \quad (2\text{-}55)$$

dengan:

P_{size} = Ukuran paket data yang terdiri dari *payload* dan *header* (byte)

ρ_b = BER (10^{-8})

b) Probabilitas Packet Loss pada Jaringan

$P_{network}$ secara umum dapat didefinisikan (Pritchard, et.al, 1993)

$$P_{network} = \frac{e^{-\frac{Eb}{No}}}{2\sqrt{\pi\left(\frac{Eb}{No}\right)}} \quad (2\text{-}56)$$

c) Signal to Noise Ratio (SNR)

Signal to Noise Ratio merupakan perbandingan antara daya yang diterima oleh penerima terhadap noise pada saat proses propagasi. Besar SNR pada skripsi ini dilihat mulai dari *transmitter* sampai ke *user*. Besarnya SNR dinyatakan dalam persamaan berikut berikut (Mischa Schwartz, 1994):

$$\text{SNR} = 10 \text{ Log } \frac{P_r}{N_0} \quad (2\text{-}57)$$

dengan:

SNR= signal to noise ratio (dB)

P_r = daya yang diterima oleh penerima (mW)

N_O = daya *noise* pada saluran transmisi (mW)

Dengan daya noise pada saluran transmisi dinyatakan dalam persamaan berikut :

$N_o = 10 \log kT + 10 \log B + NF$ (2-58)

dengan :

N_o = daya *noise* saluran transmisi (dBm)

k = konstanta Boltzman (1,38 x 10^{-23} J/K)

T = suhu *absolute* (300^o K)

NF = *noise figure*

B = *bandwidth* (Hz)

Dan dengan daya yang dapat diterima oleh penerima dapat dinyatakan dengan persamaan (Robert G. Winch, 1998: 184) :

$$P_r(dBm) = P_t - FSL - L_t - L_r + G_r + G_t \quad (2\text{-}59)$$

Sedangkan nilai FSL (*Free Space Loss*) dapat menggunakan persamaan (Andrea Goldsmith, 2005: 49) :

$$FSL = 20 \log \frac{4\pi d}{\lambda} \quad (2\text{-}60)$$

dengan:

P_r = daya terima *receiver* (dBm)

P_t = daya pancar *transmitter* (dBm)

FSL = *free space loss* (dB)

L_t = *transmitter losses* (*cable loss*) (dB)

L_r = *receiver losses* (*body loss*) (dB)

G_r = *gain receiver* (dBi)

G_t = *gain transmitter* (dBi)

λ = panjang gelombang (m)

f = frekuensi kerja sistem (Hz)

d = jarak antara *transmitter* dan *receiver* (m)

c = kecepatan cahaya (3x10^8 m/s)

d) Perhitungan *bit rate*

Bit rate pada modulasi adalah kecepatan pengiriman informasi melalui media transmisi. *Bit rate* adalah kecepatan tiap bit per sekon. *Bit rate* ini tergantung dengan jenis modulasi yang digunakan yang ditandai dengan banyaknya bit per simbol yang dikirim

pada tiap jenis modulasi. Perhitungan *bit rate* dinyatakan dengan persamaan berikut (Stallings, 2005):

$$R = 2B \times \log_2 n \quad (2\text{-}61)$$

dengan:

n = banyaknya bit pada suatu jenis modulasi

B = *bandwidth* jaringan yang dipergunakan (Hz)

R = *bit rate* (bps)

e) Energy Bit-to-Noise Ratio

Eb/No adalah suatu parameter penting dalam komunikasi digital. Parameter ini berhubungan dengan SNR yang biasanya digunakan untuk menentukan laju data digital dan sebagai ukuran mutu standar untuk kinerja sistem komunikasi digital. *Energy Bit to Noise Ratio* adalah perbandingan energi sinyal per bit terhadap noise. Eb/No ini dipengaruhi oleh SNR, dan bit rate pada modulasi.

Perhitungan Eb/No, terutama digunakan ketika akan menentukan performansi *Bit Error Rate* (BER) pada skema modulasi ditigal yang berbeda. Energi per bit dalam sebuah sinyal dijelaskan dalam persamaan berikut (William Stallings,2007):

$$\frac{Eb}{No} = \text{SNR-}10 \log \frac{B}{R} \quad (2\text{-}62)$$

dengan:

$\frac{E_b}{N_0}$ = *rasio energi bit* terhadap *noise* (dB)

SNR = *signal to noise ratio* sistem (dB)

B = *bandwidth* sistem (Hz)

R = *bit rate* (bps)

2.4.4 *Throughput*

Throughput adalah parameter yang digunakan untuk mengetahui banyaknya jumlah data yang diterima dalam keadaan baik yang ditransmisikan dari sumber data ke penerima. *Throughput* yang mungkin didapat dengan memperhatikan probabilitas paket diterima dalam keadaan salah (ρ) dapat dihitung dengan persamaan (2.61) (Mischa Schwartz, 1987).

$$\gamma = \frac{1}{t_v} = \frac{(1-\rho_{tot})}{t_1[1+(\alpha-1)\rho_{tot}]} \quad (2\text{-}63)$$

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 (Mischa Schwartz, 1987),

$$\alpha = 3 + \frac{2t_p}{t_I} \quad (2\text{-}64)$$

Waktu total yang dibutuhkan untuk melewatkan sebuah paket dirumuskan dalam persamaan sebagai berikut,

$$t_p = \frac{t_{ptot}}{N_{paket}} \quad (2\text{-}65)$$

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

$$t_1 = \frac{(PL_{frame}+H_{frame})\times 8}{C_{trans}} \quad (2\text{-}66)$$

Keterangan:

α = konstanta perbandingan

t_P = waktu total yang dibutuhkan untuk melewatkan 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.114

Menurut ITU-T G.114, kualitas suatu jaringan jika dilihat dari *delay*-nya ditampilkan sebagai berikut:

Tabel 2.6 Pengelompokan *Delay* berdasarkan ITU-T G.114

***Delay* (ms)**	**Kualitas**
0-150	Baik
150-400	Cukup, masih dapat diterima
>400	Buruk

Sumber: ITU-T G.114, 2000

2.5.2 Rekomendasi ITU-T G.1010

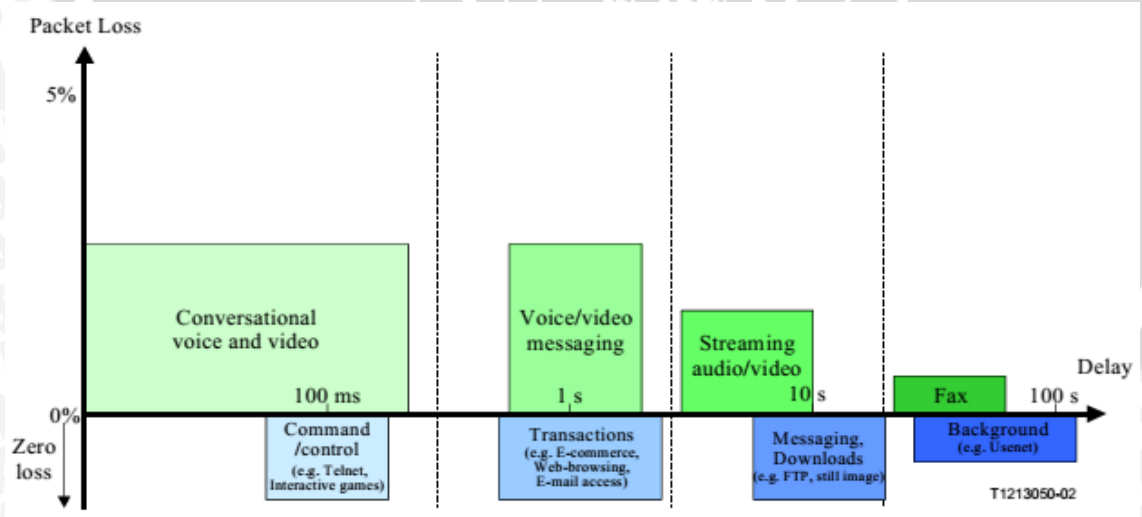
Berdasarkan ITU-T G.1010, Aplikasi *streaming* memiliki persyaratan *packet loss* yang berbeda-beda berdasarkan media yang dikirimkan. Berikut adalah tabel yang menunjukkan syarat *packet loss* berdasarkan ITU-T G.1010.

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

Medium	Application	Degree of Symmetry	Information Loss
Audio	*Conversational Voice*	*Two-way*	*< 3% Packet Loss Ratio (PLR)*
Audio	*Voice Messaging*	*One-way*	*< 3% PLR*
Audio	*High Quality Audio Streaming*	*One-way*	*< 1% PLR*
Video	*Videophone*	*Two-way*	*< 1% PLR*
Video	*Streaming*	*One-way*	*< 1% PLR*

Sumber: ITU-T G.1010, 2002

Adapun persyaratan untuk QoS parameter *delay* dan *packet loss* juga ditunjukkan pada gambar 2.13,



Gambar 2.13 *Mapping of User-Centric QoS Requirements*
Sumber: ITU-T G.1010, 2001

2.5.3 Perangkat Lunak

2.5.3.1 Wireshark

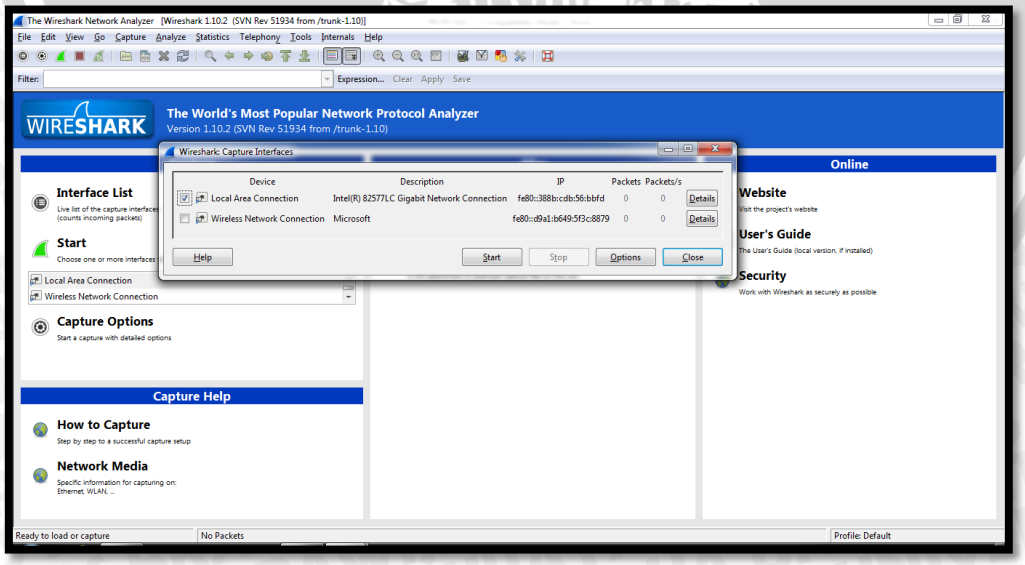
Wireshark adalah perangkat lunak *open source* dan bersifat gratis yang banyak digunakan orang-orang di dunia untuk menghitung, menganalisis paket data yang melewati suatu perangkat, umumnya komputer atau laptop.

Gambar 2.14 Logo Wireshark
Sumber: www.wireshark.org

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

- *Capture*. Wireshark menangkap paket data yang melewati adapter kabel jaringan maupun wireless pada pengguna.
- *Decode*. *Decode* atau dekode adalah proses mengubah suatu protokol menjadi sebuah informasi yang dapat diketahui.
- *Analyze*. *Analyze* membuat wireshark menjadi program paling disenangi *Network Analyst* karena dapat menampilkan beberapa parameter terkait performansi maupun karakteristik paket data.

Dan berikut adalah gambar 2.17 yang menunjukkan tampilan dari Wireshark meliputi *menubar, toolbar* dan *active sheet*.



Gambar 2.15 Lembar Kerja Wireshark
Sumber: Dokumentasi

2.5.3.2 WinPcap

WinPcap adalah perangkat tambahan bagi perangkat *packet sniffer* seperti Wireshark. Adapun fungsi dari perangkat ini, antara lain:

- WinPcap mengijinkan aplikasi seperti Wireshark untuk menangkap dan mentransmisikan paket jaringan melewati protokol-protokol.
- *kernel packet filtering*,
- mesin statistik jaringan,
- dan mendukung *remote packet capture*.

WinPcap digunakan pada banyak aplikasi, baik komersial maupun gratis: *protocol analyzers*, *network monitors*, *network intrusion detection systems*, *sniffers*, *traffic generators*, *network testers*, dan lain-lain. Beberapa dari aplikasi tersebut, seperti Wireshark, Nmap, Snort, Windump, ntop sangat sering digunakan di kalangan profesional. (winpcap.software.informer.com, 2014)

2.5.4 Standar Protokol

Header adalah bit-bit yang berisi informasi tujuan paket data dikirimkan, urutan maupun sumber paket data itu sendiri. *Header* memiliki panjang dengan satuan byte pada umumnya. Panjang *header-header* protokol yang digunakan pada penelitian ini ditunjukkan oleh tabel 2.10 sebagai berikut,

Tabel 2.8 *Header Format*

Protokol	Kuantitas	Sumber
RTP Header	12 bytes	Peterson, L, et al. 2007. Computer Networks. San Fransisco: Morgan Kaufmann: 430
Ipv4 Header	20 bytes	RFC 791
UDP Header	8 bytes	Stalling Williams. 2007. Data and Computer Communications. USA: Prentice Hall: 39
MTU Ethernet	1500 bytes	RFC 1191
GTP header	8 Byte	ETSI TS 129 060 V5.14.0
Frame Check Sequence	4 byte	IEEE 802.3
Header Ethernet	14 byte	IEEE 802.3
Header CPCS PDU	8 byte	ITU-T I.363.3
Header AAL5	5 byte	RFC 2684
Header PDCP	3 byte	ETSI TS 125 323 V3.6.0
RLC PDU size	40 byte	ETSI TS 125 322 V6.3.0, Dahlman, Erik et all. 2008. 3G Evolution HSPA and LTE for Mobile broadband. Oxford: AP
RLC Header	2 byte	ETSI TS 125 322 V6.3.0, Dahlman, Erik

		et all. 2008. 3G Evolution HSPA and LTE for Mobile broadband. Oxford: AP
ATM Header	5 byte	RFC 2684
ATM payload	48 byte	RFC 2684
Transport Block Size (TBS)	7298	Siemens LTE White Paper
MAC-d Header	4 bits	Dahlman, Erik et all. 2008. 3G Evolution HSPA and LTE for Mobile broadband. Oxford: AP
Header AAL2	3 byte	3GPP TSG-RAN Working Group 3 IP UTRAN

Sumber: Pengambilan Data Sekunder

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA