

BAB II

TINJAUAN PUSTAKA

2.1 Umum

Internet Protocol Television (IPTV) adalah sebuah sistem yang digunakan untuk mengirimkan layanan televisi digital kepada konsumen. Pengiriman sinyal digital televisi tersebut menggunakan internet protokol (IP) melewati sebuah koneksi *broadband* yang digunakan dalam sebuah jaringan dengan kualitas yang lebih baik daripada akses internet publik dengan tujuan agar kualitas pelayanan terjamin. Jaringan *broadband* yang dapat mendukung *bandwidth* tinggi untuk layanan IPTV adalah *Long Term Evolution* (LTE).

Teknologi telekomunikasi yang dikembangkan oleh 3GPP (*The 3rd Generation Partnership Project*) telah berevolusi menjadi jaringan LTE (*Long Term Evolution*) yang diperkenalkan sebagai 3GPP *release* 8. Tujuan dari evolusi yang berkelanjutan adalah untuk menyediakan komunikasi data dengan kecepatan yang tinggi dan *latency* yang rendah. Selain itu jaringan ini menggunakan *multiple access* TDD. Akses jamak ini mempunyai *bandwidth* yang tinggi dan fleksibel tergantung jenis data yang dikirim, sehingga cocok dengan layanan IPTV.

Pada bab ini akan dijelaskan teori penunjang mengenai IPTV, jaringan LTE dan *multiple access* TDD pada jaringan LTE serta parameter-parameter yang digunakan untuk mengetahui performansi IPTV pada jaringan LTE dengan mode TDD.

2.2 *Internet Protocol Television* (IPTV)

2.2.1 Definisi IPTV

Pada awal nya , IPTV disebut *Internet Protocol Television*, *Telco TV*, atau *TV broadband*, adalah suatu teknologi yang memberikan layanan televisi, *video on demand* dan konten *audio* melalui jaringan *broadband*. Definisi yang disetujui oleh *International Telecommunication Union Focus Group on IPTV* (ITU-T FG IPTV) adalah sebagai berikut:

IPTV didefinisikan sebagai layanan multimedia seperti televisi / video / *audio* / text / grafik / data yang disampaikan melalui jaringan berbasis *Internet Protocol* (IP) yang memberikan tingkat kualitas layanan, keamanan, interaktivitas dan keandalan yang tinggi.

IPTV memiliki sejumlah fitur:

- *Support TV interaktif*

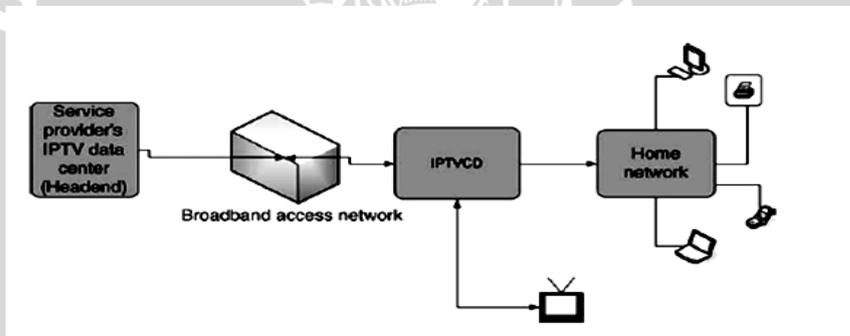
Kemampuan sistem dua arah pada IPTV yang memungkinkan penyedia layanan untuk memberikan aplikasi TV interaktif. Jenis pelayanan yang diberikan melalui layanan IPTV mencakup TV standar, *High Definition Television* (HDTV), *games* interaktif, dan *browsing* internet dengan kecepatan tinggi .

- *Time shifting*

IPTV dikombinasi dengan perekam video digital yang mempunyai konten pemrograman *Time shifting*. *Time shifting* adalah sebuah mekanisme untuk merekam dan menyimpan konten IPTV .

- *Accessible on multiple devices*

Konten IPTV tidak hanya untuk sistem televisi saja. Pengguna sering menggunakan PC atau peralatan *mobile* untuk mengakses layanan IPTV ini. (Gerard O'Driscoll, 2007)



Gambar 2.1 Diagram blok sederhana dari sistem IPTV *end-to-end*
(Sumber : Gerard O'Driscoll, 2007)

- *IPTV Data Center*

Juga dikenal sebagai "*Headend*". IPTV data center menerima konten dari sumber yang beragam termasuk dari satelit. Setelah diterima, ada beberapa komponen *hardware* yang digunakan pada IPTV data center, mulai dari *encoder* dan *server* video ke router IP dan komponen keamanan yang digunakan untuk menyiapkan konten video untuk pengiriman melalui IP. Lokasi fisik IPTV data center akan ditentukan oleh infrastruktur jaringan yang digunakan oleh penyedia layanan.

- *Jaringan Broadband*

Layanan IPTV memerlukan jaringan untuk koneksi *end to end*. Dalam pengiriman konten IPTV dalam jumlah yang besar, koneksi *end to end* meningkat secara signifikan dan kebutuhan *bandwidth* pada jaringan infrastruktur tersebut bisa sangat besar.

Kemajuan dalam teknologi jaringan untuk beberapa tahun terakhir ini memungkinkan penyedia telekomunikasi untuk memenuhi permintaan jumlah *bandwidth* yang besar. Media transmisi untuk pengiriman konten IPTV antara lain *hybrid fiber*.

- IPTVCD

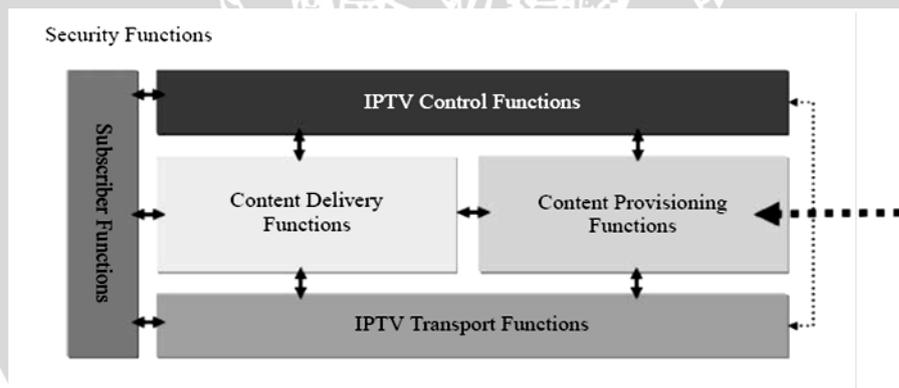
IPTV *Consumer Devices* (IPTVCD) merupakan komponen penting dalam mengakses layanan IPTV. IPTVCD terkoneksi dari jaringan *broadband* dan bertanggung jawab untuk *decoding* dan memproses video *stream* yang masuk.

- *Home Network*

Home Network menghubungkan sejumlah perangkat digital untuk lingkup yang lebih kecil. Tujuan dari *home network* adalah untuk memberikan akses informasi, seperti suara, *audio*, data, dan hiburan, dengan perangkat digital yang berbeda di sekitar rumah. Dengan *home network* pelanggan, dapat menghemat uang dan waktu karena peripheral seperti printer dan scanner, serta koneksi internet *broadband*, dapat dengan mudah dibagi. (Gerard O’Driscoll, 2007)

2.2.2 Arsitektur IPTV

IPTV dapat dibagi menjadi beberapa elemen-elemen dasar. Gambar 2.2 menunjukkan enam unsur utama dari arsitektur fungsional.



Gambar 2.2 Arsitektur Fungsional untuk layanan IPTV

(Sumber : David Ramirez, 2008)

Arsitektur fungsional dibentuk oleh enam blok utama yang mewakili fungsi: *content provision*, *content delivery*, IPTV control, IPTV transportasi, subscriber functions dan security.

1. *Content Provision*

Semua isi yang digunakan oleh layanan IPTV, termasuk VOD dan broadcasting, akan melewati fungsi *content provision* di mana fungsi pengkodean akan mempersiapkan video stream digital yang mampu didistribusikan melalui jaringan IP.

2. *Content Delivery*

Fungsi blok *content delivery* adalah bertanggung jawab untuk menyampaikan pengkodean *stream* ke pelanggan. Informasi akan diperoleh dari fungsi *IPTV control* dan *IPTV transport* untuk disampaikan ke konten pelanggan yang tepat. Fungsi *content delivery* adalah untuk pengiriman konten yang cepat dan dapat menyimpan konten seperti VOD. Ketika *subscriber functions* berinteraksi dengan *IPTV control* untuk meminta isi tertentu, mereka akan diarahkan ke fungsi *content delivery* untuk mendapatkan akses video *stream*.

3. *IPTV Control dan IPTV transport*

Fungsi *IPTV control* adalah bertanggung jawab untuk menghubungkan semua fungsi lainnya dan memastikan bahwa layanan operasional yang sesuai untuk kepuasan pelanggan. Dari sudut pandang keamanan, fungsi *IPTV control* ini penting karena bertindak sebagai *gateway* untuk konten *IPTV* yang berlangganan dan mengkoordinasikan arus data antar komponen. Fungsi *IPTV control* adalah menerima permintaan dari pelanggan dan berhubungan dengan fungsi *content delivery* dan fungsi *IPTV transport* untuk memastikan bahwa konten yang dikirim sudah sampai ke pelanggan. Fungsi tambahan dari *IPTV control* adalah untuk memberikan sebuah panduan program yang dapat digunakan oleh pelanggan untuk memilih konten yang akan disampaikan.

4. *Subscriber Functions*

Elemen yang digunakan oleh pelanggan yang memiliki akses ke konten *IPTV*. *Subscriber functions* mempunyai beberapa elemen yang bertanggung jawab untuk komunikasi dengan fungsi *IPTV transport*, misalnya sebuah web browser yang menggunakan *set top box* untuk terhubung dengan *server*. Dalam fungsi ini, *set top box* menyimpan jumlah elemen penting termasuk kunci DRM dan informasi autentifikasi pengguna. Blok pelanggan akan menggunakan EPG untuk memilih kontak yang dapat mengakses dan meminta dari fungsi *IPTV control*, selain itu pelanggan akan menerima lisensi digital dan kunci DRM untuk mengakses isinya.

5. *Security*

Semua fungsi dalam model *IPTV* didukung oleh elemen *security*. *IPTV control* dan fungsi *IPTV transport* akan mengandalkan standar *security* untuk menghindari pencampuran konten *IPTV* yang tidak sah. Keamanan pada *subscriber functions* terletak pada *set top box*. Secara umum, semua aplikasi dan sistem operasi dalam *IPTV*

harus memiliki mekanisme keamanan yang tersedia untuk menghindari kerusakan. (David Ramirez, 2008)

2.2.3 Codec IPTV

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/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*.

Beberapa *codec* yang umum digunakan dalam IPTV adalah sebagai berikut:

1. MPEG-2

Digunakan pada DVD dan video digital untuk penyiaran. *Codec* ini sedang diganti dengan versi baru. Kebanyakan program komputer menggunakannya untuk memvisualisasikan DVD, dan video internet mendukung MPEG-2 sebagai standar untuk video.

2. H.263

H.263 (ITU-T rekomendasi H.263). *Codec* ini telah diterbitkan oleh *International Telecommunication Union* (ITU) didedikasikan untuk *audiovisual* dan sistem multimedia. Rekomendasi ini meliputi kompresi gambar bergerak dengan tarif rendah bit. Rendahnya tingkat bit *output* memungkinkan untuk digunakan untuk konferensi video dan video internet. *Codec* ini memberikan peningkatan kemampuan kompresi untuk video *progressive scan* dan secara luas digunakan pada situs internet. (David Ramirez, 2008)

3. H.264

Rekomendasi ITU-T H.264 | ISO / IEC 14496-10 untuk video *coding* direkomendasikan sebagai standar video *coding* yang sangat cocok untuk digunakan dalam layanan IPTV. (ITU-T, 2009)

IPTV menggunakan standar *codec* video yang dikembangkan *International Engineering Consortium* (ISO/IEC) dan *International Telecommunication Union-the Telecommunication Standardization* (ITU-T). Standar ini juga dikenal sebagai H.264/AVC (*Advanced Video Coding*). H.264/AVC adalah standar kompresi video yang mempunyai kemampuan kompresi lebih besar dari pendahulunya bahkan jika digunakan pada *data rate* di bawah 1 Mbps. Melalui H.264/AVC, *video coding* sangatlah efisien, dalam hal ini membuat integrasi grafis yang dihasilkan computer dengan berbagai obyek yang tersedia sangatlah sederhana, lebih *fleksibel* dimaksudkan

agar menambah kemampuan untuk penyiaran video dan untuk mendukung pengembangan video digital dan efisien dalam penggunaan *bandwidth* tentunya. (ITU-T, 2009)

Tabel 2.1 Spesifikasi *Codec* H.264/AVC

Kategori kompresi IPTV	Layanan TV SD multicast (Mbps)	VOD dan SD premium (Mbps)	HDTV (Mbps)
H.264/AVC	1,75	2,1	15

(Sumber : Gerard O’Driscoll, 2007)

4. ITU-T G.719

Codec ITU-T G.719 *fullband* adalah *codec audio* yang mempunyai *bandwidth audio* mulai dari 20 Hz sampai 20 kHz dan *sample rate* 48 kHz. *Data rate* pada *codec* ini sebesar 32 kbps – 128 kbps. *Codec* G.719 adalah *codec* yang mempunyai kualitas audio yang tinggi dan rendah *complexity*. *Complexity* parameter *codec* yang dapat meningkatkan produksi daya yang dipakai. *Codec* ini cocok digunakan untuk aplikasi *video conference*, *streaming audio* dengan internet. (Polycom, 2009)

Tabel 2.2 Spesifikasi *Codec* G.719

	<i>Bandwidth</i>	<i>Bit Rate</i>	<i>Delay</i>	Standar
G.719	20 kHz	32 kbps – 128 kbps	40 ms	ITU

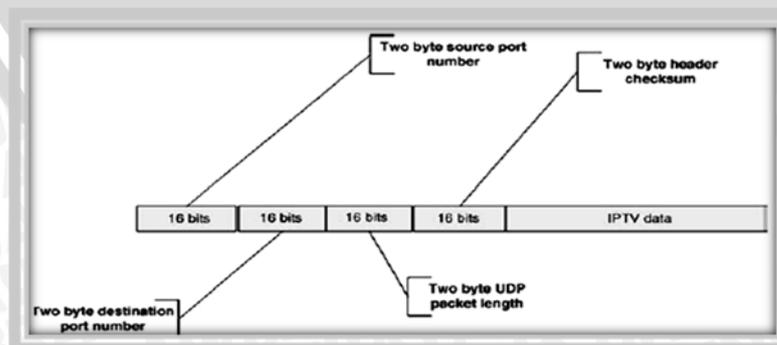
(Sumber : Polycom, 2009)

2.2.4 Protokol *Internet Protocol Television* (IPTV)

Protokol standar yang digunakan IPTV:

1. *User Datagram Protocol* (UDP)

Protokol ini adalah yang paling awal dan paling sederhana dalam teknologi transfer paket data. UDP digunakan untuk mentransfer video dan paket data yang sangat sensitif terhadap waktu transfer. Panjang *header* UDP 8 byte.



Gambar 2.3 Struktur *Header* UDP

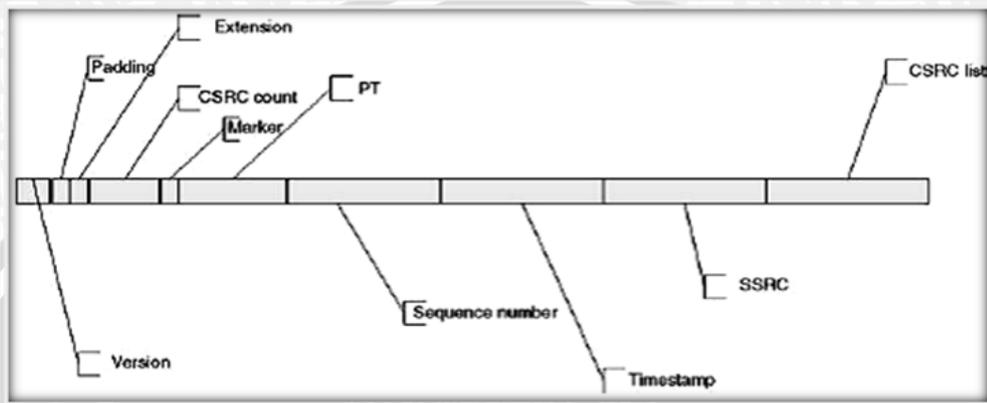
(Sumber : Gerard O’Driscoll, 2007)

2. *Transmission Control Protocol (TCP)*

Protokol yang secara umum digunakan pada teknologi transfer data. Sebagian besar perangkat yang terhubung ke internet mampu mendukung protokol ini.

3. *Real Time Protocol (RTP)*

Protokol ini dibangun secara spesifik untuk mendukung transfer paket data secara *real time* tinggi. Panjang *header* RTP 12 byte.

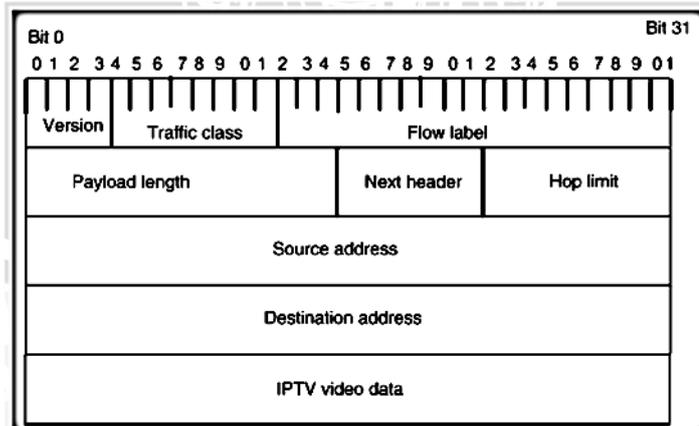


Gambar 2.4 Struktur *Header* RTP

(Sumber : Gerard O’Driscoll, 2007)

4. *Internet Protocol (IP)*

Protokol ini digunakan untuk mengirim route data sampai ke tujuan. IP disebut protokol paling baik di dalam jaringan, karena dia menyediakan layanan pengiriman paket ke seluruh layanan IPTV. Jenis IP versi 6 sekarang banyak digunakan pada layanan IPTV. Panjang *header* IP versi 6 adalah 40 byte. (Sumber : Gerard O’Driscoll, 2007)



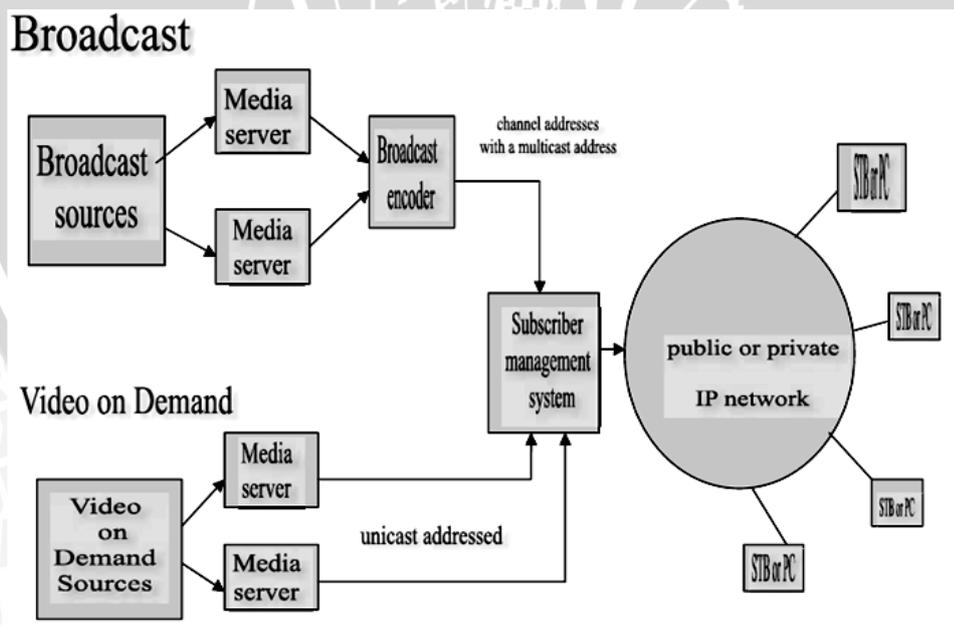
Gambar 2.5 Struktur *Header* IP

(Sumber : Gerard O’Driscoll, 2007)

2.2.5 Metode Pengiriman IPTV

Ada beberapa metode pengiriman yang digunakan pada IPTV, yaitu:

- 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.
- Multicast*, merupakan bentuk komunikasi *one to many*, tetapi tidak seperti *broadcast*. Pada *multicast*, informasi yang dikirimkan berasal dari satu sumber atau satu 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*.
- 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. (www.itelkom.ac.id/library)



Gambar 2.6 Blok Diagram Pengiriman IPTV via *Broadcasting* dan *Unicast Video On Demand*

(Sumber : Gilbert Held, 2007)

2.3 Long Term Evolution (LTE)

2.3.1 Spesifikasi Long Term Evolution (LTE)

Tujuan dari LTE adalah untuk memberikan *radioaccess* yang mempunyai kecepatan data yang tinggi, *latency* yang rendah dan paket data yang dioptimalkan, teknologi yang mendukung penyebaran *bandwidth* yang *fleksibel*. Secara paralel, arsitektur jaringan baru ini dirancang dengan tujuan untuk mendukung *packet-switched mobilitas* yang lancar, kualitas layanan dan *latency minimal*.

Sistem ini mendukung *downlink peak data rate* 326 Mbps dengan 4×4 MIMO (*Multiple Input Multiple Output*) dalam *bandwidth* 20 MHz. Karena *uplink* MIMO tidak digunakan dalam rilis pertama pada standar LTE, jadi *uplink peak data rate* terbatas untuk 86 Mbps dalam 20 MHz *bandwidth*. Dalam hal *latency*, jaringan LTE menyediakan kemampuan kurang dari 10 ms untuk transmisi paket dari jaringan untuk UE. (Farooq Khan, 2009)

Tabel 2.3 Spesifikasi Teknis untuk Teknologi LTE

Parameter Spesifikasi Teknis	Jenis/Nilai
<i>Peak data rates</i>	100 Mbit/s for downlink; 50 Mbit/s for uplink
<i>Average user throughput per MHz than HSPA Release 6</i>	3-4 higher for downlink; 2-3 higher for uplink
<i>Bandwidth</i>	1.25-20 MHz
<i>Spectrum allocation</i>	Operation in paired spectrum (FDD) and unpaired spectrum (TDD) should be supported
<i>Multiple access</i>	OFDMA (downlink) SC-FDMA (uplink)
<i>Modulation</i>	QPSK, 16-QAM, 64-QAM
<i>Latency</i>	5ms user-plane latency at IP layer, for one-way 100 ms control-plane latency from idle to active state
<i>User per cell</i>	At least 200 at 5MHz bandwidth At least 400 at bandwidth > 5MHz

(Sumber : K. Fazel and S. Kaiser, 2008)

2.3.2 Arsitektur Long Term Evolution (LTE)

Arsitektur jaringan LTE dirancang dengan tujuan untuk mendukung *packet-switched* dengan *mobilitas* yang lancar, kualitas layanan (QoS) dan *latency minimal*. Sebuah *packet-switched* ini memungkinkan untuk mendukung semua layanan termasuk

suara melalui koneksi paket. Arsitektur ini disederhanakan dengan hanya mempunyai dua jenis elemen yaitu *evolved Node-B* (eNB) dan *Mobility Management Entity/Gateway* (MME/GW).

Hal ini berbeda dengan banyaknya jaringan *node* dalam arsitektur jaringan 3G. Salah satu perubahan yang paling utama adalah *Radio Network Controller* (RNC) dihilangkan dari arsitektur jaringan dan fungsinya kini tergabung dalam eNB. Beberapa manfaat dari sebuah *node* tunggal dalam jaringan akses adalah berkurangnya *latency* dan berkurangnya distribusi beban pengolahan RNC ke beberapa eNB. (Farooq Khan, 2009)

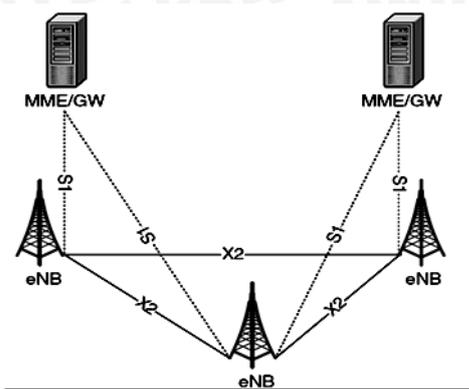
Semua *interface* jaringan didasarkan pada protokol IP. eNB saling berhubungan dengan *interface* X2 dan entitas MME / GW melalui *interface* S1 seperti yang ditunjukkan pada Gambar 8. *Interface* S1 mendukung hubungan antara MME / GW dan eNB. (Farooq Khan, 2009)

eNB dihubungkan dengan *interface* S1 MME ke *Mobility Manajemen Entity* (MME) dan *Service Gateway* (S-GW) dengan *interface* S1-U. (Luca Reggiani, 2009)

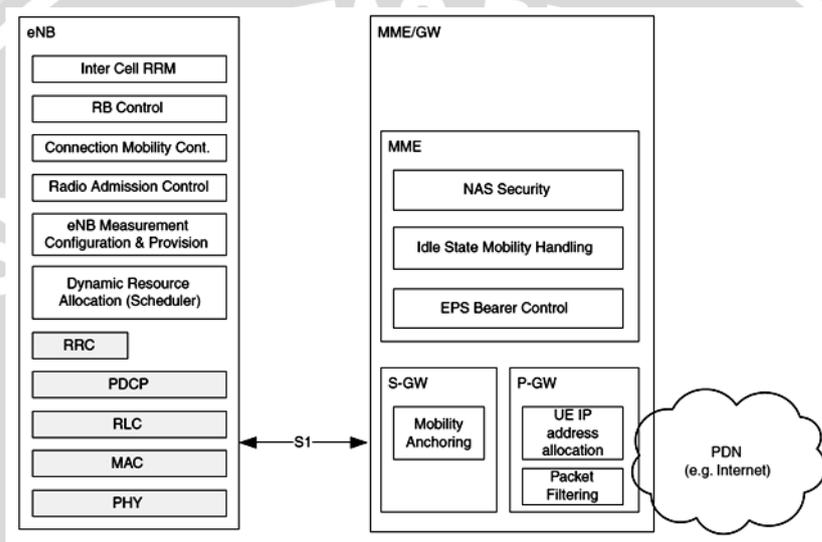
Perpecahan fungsi antara eNB dan MME / GW ditunjukkan pada Gambar 9. *Evolved Packet Core Network* (EPC) terdiri dari *gateway* pada jaringan LTE dan MME. *Gateway* pada LTE dibagi menjadi *Service Gateway* (S-GW) dan *Packet Data Network Gateway* (P-GW). S-GW bertindak sebagai pembawa *mobilitas* lokal yang dapat meneruskan dan menerima paket ke dan dari eNB untuk UE. P-GW adalah *interface* dengan *eksternal Packet Data Network* (PDN) seperti internet dan IMS. P-GW juga melakukan beberapa fungsi IP seperti alokasi alamat, paket *filtering* dan *routing*.

MME digunakan untuk pensinyalan dan karenanya pengguna paket IP tidak melalui MME. Sebuah keuntungan dari entitas jaringan yang terpisah dari pensinyalan adalah kapasitas jaringan untuk sinyal dan trafik dapat tumbuh secara mandiri.

Evolved Node-B mengimplementasikan fungsi Node-B serta protokol yang diimplementasikan dalam RNC. eNode-B bertindak sebagai jembatan antara UE dan EPC. Fungsi utama dari eNB adalah kompresi/dekompresi *header Internet Protocol* (IP), yang berarti menghindari berulang kali mengirimkan data yang sama atau berurutan dalam *header* IP. Beberapa manfaat dari sebuah *node* tunggal di jaringan akses adalah *latency* berkurang dan distribusi beban RNC diproses menjadi beberapa eNB. (Farooq Khan, 2009)

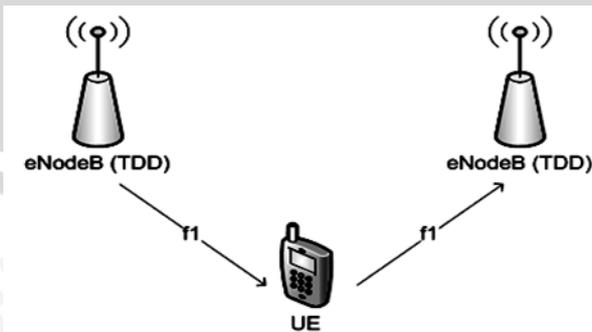


Gambar 2.7 Arsitektur Sederhana LTE
(Sumber : Farooq Khan, 2009)



Gambar 2.8 Fungsi antara MME/GW dengan eNB
(Sumber : Farooq Khan, 2009)

Pada interface ENodeB dengan *user* memungkinkan untuk transfer data hingga 100 Mbps. Operasi mode *Time Division Duplex* (TDD) digunakan karena operasi ini dapat memenuhi kebutuhan pasar dalam pelayanan transmisi data yang tinggi.

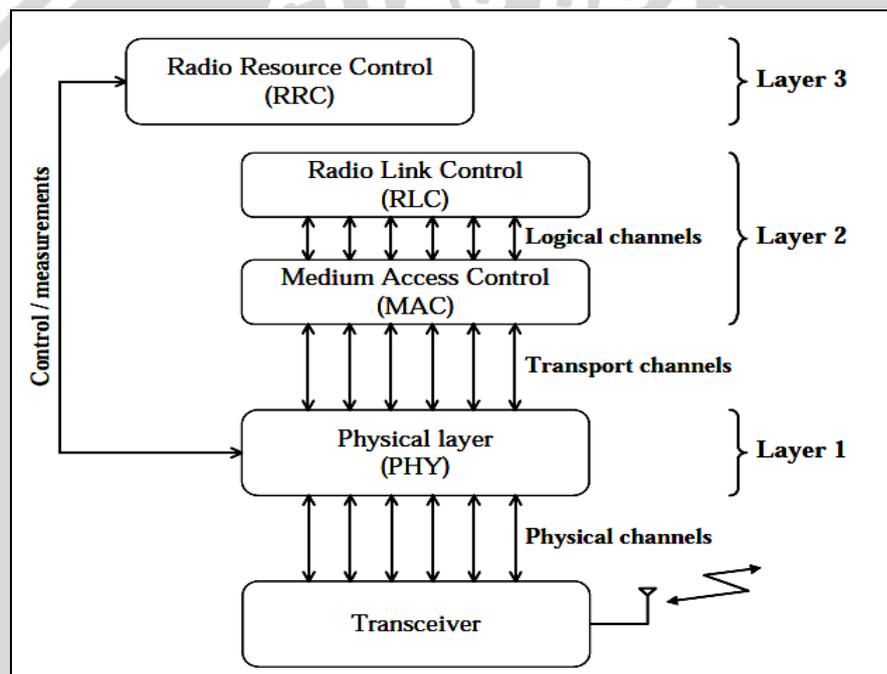


Gambar 2.9 Operasi Mode *Time Division Duplex* (TDD)
(Sumber : H. Holma dan A. Toskala, 2009)

Sistem seluler LTE memiliki spektrum frekuensi *unpaired band* yang digunakan untuk menangani trafik asimetris. Pengertian *unpaired band* disini adalah suatu band spektrum frekuensi yang memiliki frekuensi pembawa tidak sepasang dengan frekuensi pembawa pada bidang frekuensi yang lain, atau dengan kata lain frekuensi yang digunakan untuk transmisi *uplink* dan *downlink* adalah sama.

2.3.3 Protokol Long Term Evolution (LTE)

Ketika paket data dikirim melalui jaringan LTE, paket data akan melewati node yang mempunyai beberapa protokol atau aturan agar paket data itu dapat terarah ke tempat tujuan dengan benar. Dibawah ini adalah protokol yang digunakan di tiap node pada jaringan LTE.



Gambar 2.10 Protokol LTE

(Sumber : Hyung G. Myung, 2009)

Fungsi protokol di atas 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 mengompres *header IP*, pengiriman data dari pengguna. Selain itu, PDCP bertanggung jawab dalam fungsi pengendalian pengiriman data dan *ciphering*. Panjang *header PDCP* 2 byte.



c. *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. 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. *Layer Fisik (PHY)*

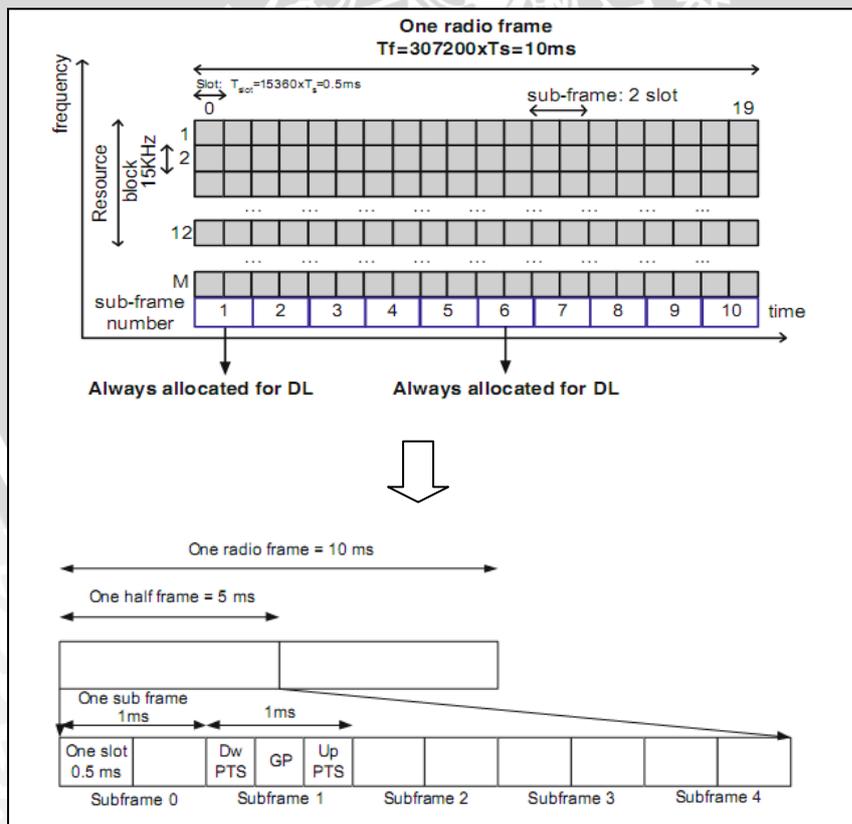
Fungsi utama dari layer fisik adalah *coding*, modulasi dan transmisi *multiple* antena (MIMO).

f. *GPRS Tunnelling Protocol (GTP)*

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

(K. Fazel dan S. Kaiser, 2008)

Selain itu, ada struktur *frame* pada jaringan LTE :



Gambar 2.11 *Frame* LTE

(Sumber : Mustafa Ergen, 2009)

2.3.4 Kelebihan *Long Term Evolution* (LTE)

1. Pada jaringan HSPA saat ini, waktu yang dibutuhkan ponsel perangkat untuk terhubung ke jaringan dan memulai komunikasi pada *user* relatif panjang. Ini memiliki dampak negatif pada kegunaan, sebagai pengguna bisa merasakan penundaan ini ketika mengakses layanan di Internet setelah beberapa waktu ketika tidak aktif. Dengan itu, diputuskan bahwa dengan desain jaringan baru itu harus memungkinkan untuk berpindah dari keadaan *idle* menjadi sepenuhnya terhubung dalam waktu kurang dari 100 ms.

2. Mengurangi *latency user* - Kelemahan dari jaringan selular saat ini adalah keterlambatan transmisi jauh lebih tinggi dibandingkan dengan jaringan telepon tetap. Sementara penundaan satu arah antara komputer pengguna di jaringan *edge* DSL ke Internet adalah sekitar 15 ms, jaringan HSPA mengalami keterlambatan sekitar 50 ms. Ini tidak menguntungkan untuk aplikasi seperti telepon dan *game real-time*. Untuk pada LTE, penundaan *air interface* harus berada di urutan 5 ms untuk mencapai *end-to-end*.

3. Meningkatkan *throughput* - untuk sistem LTE, *throughput* maksimal yang ideal pada kondisi 100 Mbps harus dicapai. (Martin Sauter, 2008)

2.3.5 *Time Division Duplex* (TDD)

Dalam suatu komunikasi radio dimungkinkan penggunaan secara bersama suatu *channel* berdasarkan pembagian yang dilakukan pada domain waktu. Atas dasar pemikiran inilah yang membuat TDD dapat digunakan sebagai metode full duplex dalam menyelenggarakan suatu komunikasi dua arah yang bersifat simultan. Masing-masing pengguna mempunyai dua *channel* yaitu *forward* dan *reverse* yang terbentuk dari alokasi slot-slot waktu, sehingga TDD mengizinkan dua *channel* terletak pada band frekuensi yang sama. Suatu slot waktu akan dipisahkan untuk digunakan sebagai *channel forward* dan *channel reverse*. Dengan demikian dua keperluan yang berbeda yaitu transmisi dan penerimaan sinyal dapat ditangani oleh dua *channel*.

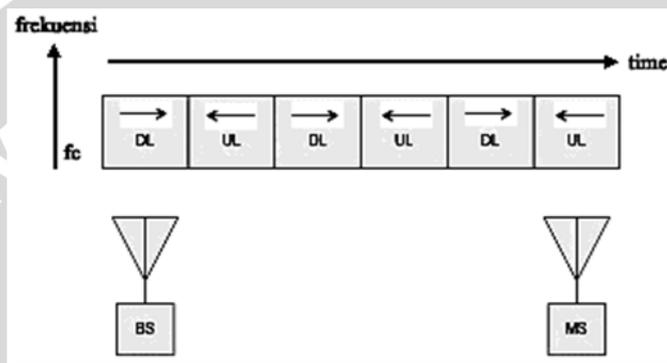
Metode *duplex* TDD sangat cocok untuk diaplikasikan pada jaringan sel yang kecil. Ukuran sel tersebut adalah pikosel hingga mikrosel. TDD merupakan suatu solusi komunikasi *full duplex* dengan biaya rendah dan dapat menghemat penggunaan dari frekuensi. TDD pun dapat melayani transmisi layanan suara dan aplikasi data.

TDD memanfaatkan spektrum yang lebih efisien daripada FDD. FDD tidak dapat digunakan dalam lingkungan di mana penyedia layanan tidak memiliki cukup

untuk menyediakan *bandwidth* yang diperlukan antara *guardband* pengiriman dan saluran penerima.

2.3.5.1 Prinsip Transmisi TDD

TDD dapat bekerja pada frekuensi *uplink* dan *downlink* yang sama tetapi dialokasikan ke dalam slot-slot waktu yang berbeda dan saling dipisahkan oleh selang waktu (*guard time*). Pesan yang disampaikan pada TDD dikirimkan menggunakan frekuensi pembawa yang sama, sehingga pembagian arah transmisi lebih cepat, mudah, dan efisien. (Dixon, 1994)



Gambar 2.12 Blok Transmisi TDD

(Sumber : Dixon, 1994)

2.3.5.2 Karakteristik TDD

Karakteristik utama dari TDD adalah sistem dapat bekerja pada satu frekuensi pembawa untuk mengirim dan menerima data sehingga ideal digunakan untuk aplikasi layanan di pikosel dengan trafik asimetris. TDD dapat melayani kecepatan data sampai dengan 2 Mbps pada pengguna di dalam gedung dengan trafik yang tidak begitu padat. Prinsip pembagian kanal pada TDD merupakan kombinasi dari TDMA dan CDMA (TD-CDMA). (Holma dan Toskala, 2000)

TDD menggunakan teknik *Slow Frequency Hopping Spread Spectrum* yang dikombinasikan dengan TDMA sehingga lebih dikenal sebagai sistem *hybrid SFH TDMA/CDMA*. Sistem ini memiliki keuntungan dari TDMA atau SFH-CDMA murni, karena terjadi kolaborasi antara TDMA dan SFH-CDMA, yaitu TDMA disusun berdasarkan atas prinsip *bandwidth on demand* jadi sistem ini` pengaksesannya diatur dengan membagi *bandwidth* ke dalam slot-slot waktu untuk masing-masing pemakai. Sedangkan SFH-CDMA disusun dengan akses simultan antar sel *hybrid* sehingga dalam sebuah sel tidak diizinkan pemakaian frekuensi sama.

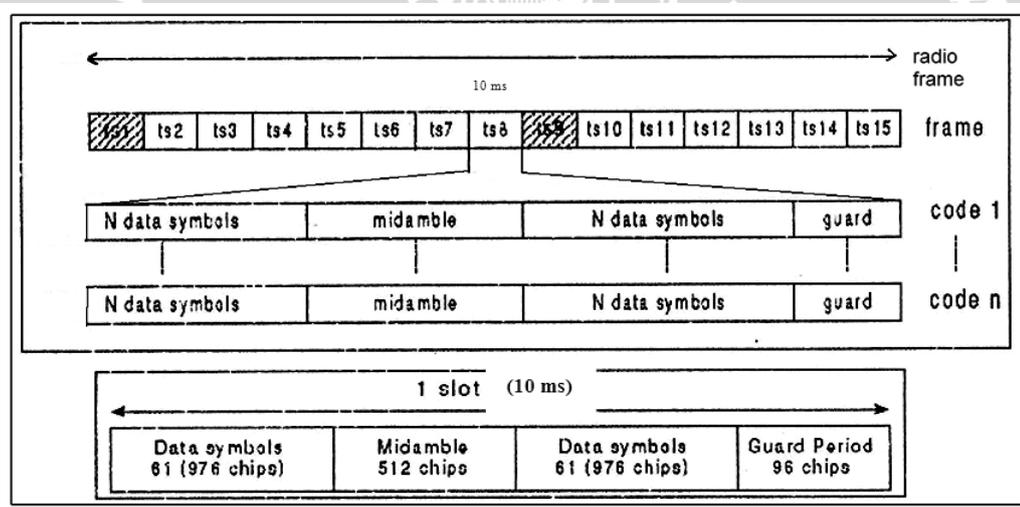
Berdasarkan penjelasan diatas, masing-masing sel pada TDD akan ditempati oleh pemakai dengan kode-kode penyebaran secara berbeda dan pada sel terdapat beberapa pemakai dengan frekuensi pembawa yang berbeda. Pemakai tersebut dialokasikan secara khusus ke dalam suatu slot waktu sehingga setiap saat hanya ada satu pemakai dengan frekuensi tertentu saja yang ditransmisikan setiap sel. (Ravi, 2000)

2.3.5.3 Struktur Kanal

Struktur kanal TDD dibagi menjadi dua, yaitu kanal fisik dan kanal pembawa.

a. Kanal Fisik

Kanal fisik TDD terdiri dari frekuensi pembawa, kode akses, slot waktu. Satu *frame* pada TDD memiliki durasi waktu 10 ms, dan setiap slot waktu mempunyai 2560 chip dengan kecepatan 3,84 Mcps. Setiap slot waktu mempunyai struktur dan tipe data *burst* yang terdiri dari n data simbol, *midamble*, dan selang waktu (*guard time*) seperti yang ditunjukkan Gambar dibawah ini



Gambar 2.13 Kanal Fisik TDD

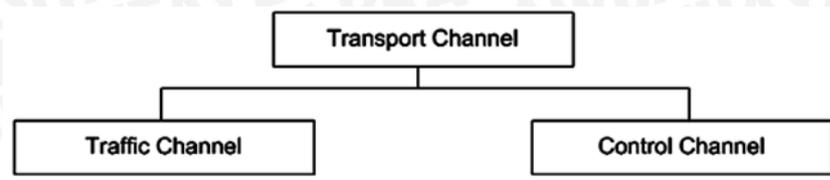
(Sumber : 3GPP, TS 25.211, 2001)

Setiap UE dapat menerima sinyal informasi pada suatu sel yang terdiri dari frekuensi, *chip*, slot waktu, sinkronisasi *frame* dan determinasi hubungan ke eNodeB.

b. Kanal Pembawa

Kanal pembawa digunakan untuk mengontrol data, memproses panggilan, dan mensinkronisasi data. Kanal pembawa dibagi menjadi dua tipe, yaitu *Traffic Channel* (THC) dan *Control Channel* (CCH) seperti ditunjukkan pada Gambar 2.13





Gambar 2.14 Kanal Pembawa TDD
(Sumber : 3GPP, TS 25.211, 2001)

a. *Traffic Channel* (TCH)

TCH digunakan untuk mentransfer data, mengontrol *signalling* data yang berisi informasi pemrosesan panggilan, serta membawa sinkronisasi dan informasi transmisi radio.

b. *Control Channel* (CCH)

Kanal CCH pada TDD digunakan untuk koneksi hubungan antar pemakai.

2.3.5.4 Spektrum LTE TDD

Spektrum frekuensi yang digunakan pada LTE TDD :

Tabel 2.4 Spektrum frekuensi yang digunakan pada LTE TDD

E-UTRA Operating Band	Uplink (UL) operating band BS receive UE transmit	Downlink (DL) operating band BS transmit UE receive	Duplex Mode
	$F_{UL_low} - F_{UL_high}$	$F_{DL_low} - F_{DL_high}$	
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
41	2496 MHz – 2690 MHz	2496 MHz – 2690 MHz	TDD
42	3400 MHz – 3600 MHz	3400 MHz – 3600 MHz	TDD
43	3600 MHz – 3800 MHz	3600 MHz – 3800 MHz	TDD
44	703 MHz – 803 MHz	703 MHz – 803 MHz	TDD

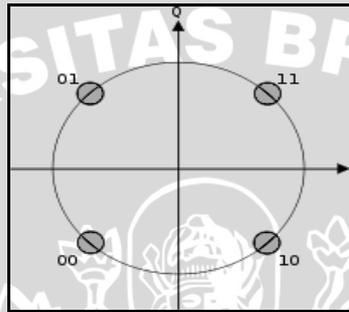
(Sumber : Bong Youl, 2012)

2.3.6 Modulasi pada *Long Term Evolution* (LTE)

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

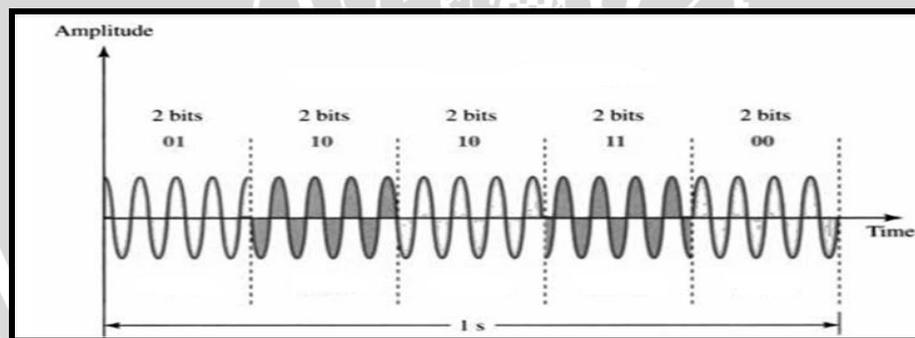
1. Modulasi QPSK

QPSK dikenal juga dengan *quaternary* atau *quadriphase* PSK atau 4 PSK. QPSK menggunakan 4 titik diagram *konstelasi* dengan mengirimkan 2 bit untuk setiap simbol (00, 01, 10, 11). Dengan menggunakan 4 fasa, QPSK dapat melakukan *encode* dua 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 dua kali lebih cepat dibandingkan dengan BPSK dan lebih efisien dalam penggunaan spektrum frekuensi. Sedangkan kelemahannya adalah kurang tahan terhadap *noise* dan rawan terjadi interferensi dari sinyal lain.



Gambar 2.15 4 Titik Diagram Konstelasi pada QPSK

(Sumber : Fuqin Xiong, 2006)



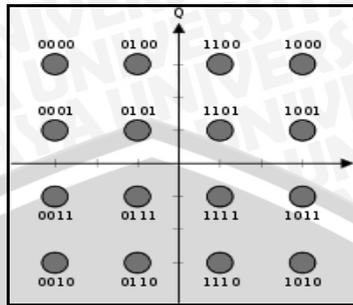
Gambar 2.16 Sinyal dengan Modulasi QPSK

(Sumber : Behrouz A. Forouzan, 2000)

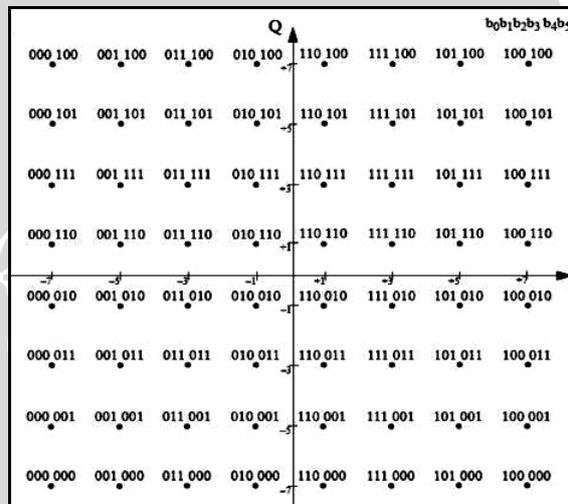
2. Quadrature Amplitude Modulation (QAM)

QAM merupakan teknik modulasi yang merupakan perpaduan antara ASK dan PSK. Sehingga pada QAM, amplitudo dan fasa sinyal *carrier* akan berubah 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, amplitudo dua sinyal *carrier* QAM yang berbeda fasa 90 derajat antara satu dengan lainnya

dirubah untuk mendapatkan sinyal yang diinginkan. Jenis modulasi QAM yang digunakan dalam teknologi LTE adalah 16-QAM dan 64-QAM.



Gambar 2.17 Diagram konstelasi 16 QAM
(Sumber : Fuqin Xiong, 2006)



Gambar 2.18 Diagram konstelasi 64 QAM
(Sumber : Fuqin Xiong, 2006)

Dalam sebuah sinyal QAM, terdapat dua operator, masing-masing memiliki frekuensi yang sama tetapi berbeda dalam tahap 90 derajat atau satu seperempat dari sebuah siklus, dari hal tersebut maka timbul istilah *quadrature*. Satu sinyal disebut sinyal I, dan sinyal yang lain dinamakan sinyal T. Salah satu sinyal dapat diwakili oleh sebuah gelombang sinus, dan sinyal lainnya dengan gelombang kosinus. Kedua sinyal tersebut digabungkan pada sumber untuk transmisi. Di tempat tujuan, sinyal informasi dan pembawa dipisahkan, sinyal informasi diambil dari masing-masing sinyal pembawa, dan kemudian digabungkan menjadi satu sinyal. (Fuqin Xiong, 2006)

2.4 Performansi IPTV pada jaringan *Long Term Evolution* (LTE) dengan mode *Time Division Duplex* (TDD)

2.4.1 *Bandwidth* Aplikasi IPTV

Tiap paket video atau audio pada aplikasi IPTV yang akan dikirim terdiri dari 2 bagian, yaitu *header* dan *payload*. *Header* berisi bit bit untuk pengalamatan sesuai dengan tujuan, sedangkan *payload* berisi bit bit data informasi yang akan disampaikan ke *user*. *Header* yang digunakan pada IPTV adalah *header* RTP, UDP dan IP. Data pertama akan masuk ke protokol RTP, dan *header* RTP digunakan pada IPTV karena pada IPTV membutuhkan pengiriman data secara *real time*, dan *header* ini dapat mendukung itu. Panjang *header* RTP adalah 14 byte. Setelah dari protokol RTP, data akan masuk ke protokol UDP, dan *header* UDP digunakan pada IPTV karena *header* ini dapat mendukung *no pause in the delivery IPTV content* dan *speed connection setup*, jadi dengan *header* ini ketika pengiriman *content* IPTV akan memperkecil *delay* yang terjadi dan kecepatan koneksinya hanya membutuhkan waktu yang pendek. Panjang *header* UDP adalah 8 byte. Dan data IPTV akan masuk ke protokol IP, dan *header* IP ini berfungsi untuk memilih jalur *routing* untuk pengiriman data. Panjang *header* IP adalah 40 byte.

Pada skripsi ini akan mengirimkan aplikasi IPTV yang terdiri dari paket IP video dan *audio*. Paket video dan *audio* ini akan dikirimkan secara bersama dalam satu paket IPTV ke beberapa *user*. Jumlah *bandwidth* yang diperlukan dalam aplikasi IPTV sangat bergantung pada *data rate codec*, *frame rate* dan jenis *codec* yang digunakan. *Data rate codec* adalah kecepatan untuk pembentukan data pada *codec* dan *frame rate* adalah waktu yang dibutuhkan untuk membentuk satu *frame* data. Untuk menghitung kebutuhan *bandwidth* pada IPTV, pada skripsi ini menggunakan jenis *codec* video H.264/AVC dengan *data rate* sebesar 1750 kbps dan *frame rate* 33 ms untuk jenis layanan televisi SD *multicast*, dan jenis *code audio* G.719 dengan *data rate* sebesar 40 kbps - 128 kbps dan *frame rate* 33 ms. Besarnya *bandwidth* pada IPTV dapat dihitung dengan persamaan 2.1:

$$B_t = B_a + B_v \quad (2.1)$$

dengan:

B_t : *Bandwidth* total (bps).

B_a : *Bandwidth* audio (bps).

B_v : *Bandwidth* video (bps).

Bandwidth total IPTV adalah jumlah *bandwidth* yang dibutuhkan IPTV untuk mengirimkan aplikasi IPTV yang berupa paket video dan *audio* atau dapat dianalogikan sebagai lebar ruang yang dibutuhkan untuk tempat mengirimkan aplikasi IPTV yang berupa paket audio dan video.

a. Paket Video

Besarnya *bandwidth* video adalah lebar frekuensi yang dibutuhkan untuk dapat mengirimkan paket video dan dapat dihitung dengan persamaan :

$$B_v = P_{size V} \times P_{vt} \tag{2.2}$$

dengan:

- B_v : *Bandwidth* video (bps).
- $P_{size V}$: Besar tiap paket video (bit/paket).
- P_{vt} : Jumlah paket video yang dihasilkan tiap detik (paket/detik).

Payload pada video menyatakan banyaknya bit-bit video yang terisi didalam paket.

Payload pada tiap paket video dapat diperoleh sesuai persamaan 2.3

$$X_{LV} = FR \times B_{codec V} \tag{2.3}$$

dengan:

- X_{LV} : *Payload* tiap paket video (bit).
- $B_{codec V}$: *Bandwidth codec* video (bps).
- FR (*Frame Rate*) : Waktu pembentukan tiap frame (s).

Besar tiap paket video adalah jumlah bit total dari penjumlahan bit pada header dan *payload* pada paket video dan dapat diperoleh sesuai dengan persamaan 2.4

$$P_{size V} = X_{LV} + header \tag{2.4}$$

Jumlah paket video yang dihasilkan tiap detik menyatakan banyaknya paket yang diperlukan untuk ditransmisikan setiap detik sesuai dengan persamaan 2.5

$$P_{vt} = B_{codec V} : X_{LV} \tag{2.5}$$

dengan:

- $P_{size V}$: Besar tiap paket video (bit/paket).
- header* : Total *header* tiap paket (60 byte).
- P_{vt} : Jumlah paket video yang dihasilkan tiap detik (paket/detik).

b. Paket Audio

Besarnya *bandwidth* audio adalah lebar frekuensi yang dibutuhkan untuk dapat mengirimkan paket audio dan dapat dihitung dengan persamaan :

$$B_a = P_{size A} \times P_{at} \tag{2.6}$$



dengan:

B_a : *Bandwidth audio* (bps).

$P_{size A}$: Besar tiap paket *audio* (bit/paket).

Pat : Jumlah paket *audio* yang dihasilkan tiap detik (paket/detik).

Payload pada audio menyatakan banyaknya bit-bit audio yang terisi didalam paket.

Payload pada tiap paket audio dapat diperoleh sesuai persamaan 2.7

$$X_{LA} = FR \times B_{codecA} \quad (2.7)$$

dengan:

X_{LA} : *Payload* tiap paket audio (bit).

B_{codecA} : *Bandwidth codec* audio (bps).

FR (*Frame Rate*) : Waktu pembentukan tiap frame (s).

Besar tiap paket audio adalah jumlah bit total dari penjumlahan bit pada header dan *payload* pada paket audio dan dapat diperoleh sesuai dengan persamaan 2.8

$$P_{size A} = X_{LA} + header \quad (2.8)$$

Jumlah paket audio yang dihasilkan tiap detik menyatakan banyaknya paket yang diperlukan untuk ditransmisikan setiap detik sesuai dengan persamaan 2.9

$$Pat = B_{codecA} : X_{LA} \quad (2.9)$$

dengan:

$P_{size A}$: Besar tiap paket *audio* (bit/paket).

Pat : Jumlah paket *audio* yang dihasilkan tiap detik (paket/detik).

$header$: Total *header* tiap paket (60 byte)

2.4.2 *Signal to Noise Ratio*

Signal to Noise Ratio merupakan perbandingan antara daya yang diterima oleh penerima terhadap noise yang timbul 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) :

$$SNR = 10 \text{ Log } \frac{P_r}{N_o} \quad (2.10)$$

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 :

$$N_o = 10 \log kT + 10 \log B + NF \quad (2.11)$$

dengan :

N_o = daya noise saluran transmisi (dBm)

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

T = suhu absolute (300° K)

NF = noise figure

B = bandwidth (Hz)

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

$$P_r(\text{dBm}) = P_t - FSL - L_t - L_r + G_r + G_t \quad (2.12)$$

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

$$FSL = 20 \log \frac{4\pi d}{\lambda} \quad (2.13)$$

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 (3×10^8 m/s)

2.4.3 Bit Error Rate (BER)

Bit Error Rate atau yang disebut juga probabilitas bit salah adalah peluang besarnya bit salah yang mungkin terjadi selama proses pengiriman data. Probabilitas bit salah pada jaringan LTE maksimal sebesar 10^{-3} . Probabilitas bit salah ini dipengaruhi oleh jenis modulasi, *bit rate* pada modulasi yang digunakan, dan *Energy Bit to Noise Ratio* (Eb/No) pada sistem. Jenis modulasi yang digunakan pada skripsi ini adalah

QPSK, 16 QAM dan 64 QAM. Jenis modulasi ini adalah jenis modulasi yang digunakan pada jaringan LTE sesuai spesifikasi jaringan LTE.

2.4.3.1 Bit Rate pada Modulasi

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):

$$R = 2B \times \log_2 n \quad (2.14)$$

dengan:

n : banyaknya bit pada suatu jenis modulasi

B : *bandwidth* jaringan yang dipergunakan (Hz)

R : *bit rate* (bps)

2.4.3.2 Energy Bit-to-Noise Ratio

E_b/N_o 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. E_b/N_o ini dipengaruhi oleh SNR, dan bit rate pada modulasi.

Perhitungan E_b/N_o , terutama digunakan ketika akan menentukan performansi *Bit Error Rate* (BER) pada skema modulasi digital yang berbeda. Energi per bit dalam sebuah sinyal dijelaskan dalam persamaan berikut (William Stallings, 2007):

$$\frac{E_b}{N_o} = SNR - 10 \log \frac{B}{R} \quad (2.15)$$

dengan:

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

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

B = *bandwidth* sistem (Hz)

R = *bit rate* (bps)

2.4.3.3 Nilai dari probabilitas bit salah dapat diperoleh dari perhitungan yang dilakukan berdasarkan modulasi sinyal yang digunakan.

a. Modulasi QPSK

Pada modulasi QPSK (*Quadrature Phase Shift Keying*) terdapat 2 bit dalam tiap simbolnya. Simbol itu dinamakan baud. Besarnya nilai probabilitas bit salah/ *bit error rate* pada modulasi QPSK adalah (Mischa Schwartz, 1994):

$$P_{b,QPSK} = Q \left(\sqrt{2 \frac{E_b}{N_0}} \right)$$

dengan $Q = \frac{1}{2} \operatorname{erfc} \frac{z}{\sqrt{2}}$

$$P_{b,QPSK} = \frac{1}{2} \operatorname{erfc} \left(\sqrt{\frac{E_b}{N_0}} \right) \quad (2.16)$$

dengan:

$\frac{E_b}{N_0}$: *energy bit to noise* (dB)

$P_{b,QPSK}$: probabilitas bit salah

b. Modulasi 16 QAM

Modulasi 16 QAM merupakan salah satu jenis dari modulasi QAM (*Quadrature Amplitude Modulation*). QAM adalah teknik modulasi yang mengkombinasikan antara teknik *Amplitude Shift Keying* (ASK) dan teknik *Phase Shift Keying* (PSK). Pada QAM, amplitudo dan *fasa* sinyal *carrier* berubah terhadap perubahan amplitudo sinyal informasi. Pada modulasi ini terdapat 4 bit tiap simbolnya.

Besarnya nilai probabilitas bit salah pada modulasi 16 QAM adalah (Mischa Schwartz, 1994):

$$P_{b,M-QAM} = \frac{4(\sqrt{M} - 1)}{\sqrt{M \log_2(M)}} \times Q \times \left(\sqrt{\frac{3 \log_2(M)}{(M - 1)} \times \frac{E_b}{N_0}} \right)$$

dengan $Q = \frac{1}{2} \operatorname{erfc} \frac{z}{\sqrt{2}}$

$$P_{b,M-QAM} = \frac{2(\sqrt{M}-1)}{\sqrt{M \log_2(M)}} \times \operatorname{erfc} \times \left(\sqrt{\frac{3 \log_2(M)}{2(M-1)} \times \frac{E_b}{N_0}} \right) \quad (2.17)$$

dengan :

$\frac{E_b}{N_0}$: *energy bit to noise* (dB)

$P_{b,M-QAM}$: probabilitas bit salah pada modulasi QAM

M : jumlah sinyal, untuk modulasi 16-QAM nilai $M = 2^4 = 16$

c. Modulasi 64 QAM

Modulasi 64 QAM merupakan jenis modulasi QAM (*Quadrature Amplitude Modulation*) yang sedikit berbeda dari modulasi 16 QAM. Pada modulasi ini terdapat kombinasi 6 bit yang berbeda pada tiap simbolnya. Teknik 64-QAM merupakan teknik yang paling cepat mentransmisikan data pada teknologi LTE, karena jumlah bit per simbolnya adalah yang tertinggi dibandingkan dengan teknik modulasi digital yang lain, yaitu 6 bit.. Besarnya nilai probabilitas bit salah pada modulasi 64 QAM adalah (Mischa Schwartz, 1994):

$$P_{b,M-QAM} = \frac{4(\sqrt{M}-1)}{\sqrt{M}\log_2(M)} \times Q \times \left(\sqrt{\frac{3\log_2(M)}{(M-1)} \times \frac{E_b}{N_o}} \right)$$

dengan $Q = \frac{1}{2} \operatorname{erfc} \frac{x}{\sqrt{2}}$

$$P_{b,M-QAM} = \frac{2(\sqrt{M}-1)}{\sqrt{M}\log_2(M)} \times \operatorname{erfc} \times \left(\sqrt{\frac{3\log_2(M)}{2(M-1)} \times \frac{E_b}{N_o}} \right) \quad (2.18)$$

dengan:

$\frac{E_b}{N_o}$: energy bit to noise (dB)

$P_{b,M-QAM}$: probabilitas bit salah pada modulasi QAM

M : jumlah sinyal, untuk modulasi 64-QAM nilai $M = 2^6 = 64$

Nilai $\operatorname{erfc}(x)$ atau *complementary error function* (x) didefinisikan sebagai fungsi kesalahan dari variabel (x) dapat dinyatakan sebagai:

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-y^2} dy$$

Jika $u = -y^2$, $du = -2y dy$ maka $dy = \frac{du}{-2y}$

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^u dy$$

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^u \frac{du}{-2y}$$

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \frac{1}{-2y} \int_x^{\infty} e^u du$$

$$\operatorname{erfc}(x) = \sum_x^{\infty} -\frac{1}{\sqrt{\pi}y} e^u$$

$$\begin{aligned}
 \operatorname{erfc}(x) &= \sum_x^{\infty} -\frac{1}{\sqrt{\pi}y} e^{-y^2} \\
 \operatorname{erfc}(x) &= 0 - \left(-\frac{1}{\sqrt{\pi}x} e^{-x^2}\right) \\
 \operatorname{erfc}(x) &= \frac{1}{\sqrt{\pi}x} e^{-x^2} \\
 \operatorname{erfc}(x) &\approx \frac{1}{\sqrt{\pi}x} e^{-x^2} \tag{2.19}
 \end{aligned}$$

dengan,

erfc = *error function complementary*

2.4.4 Probabilitas *Packet Loss* IPTV pada Jaringan LTE dengan Mode *Duplex* TDD

Packet loss adalah jumlah paket yang hilang pada waktu pentransmisian dibandingkan dengan jumlah paket yang dikirim. Pada skripsi ini *packet loss* adalah paket yang hilang pada waktu pentransmisian. Ini dikarenakan probabilitas *packet loss* dipengaruhi oleh besarnya probabilitas bit salah yang terjadi pada waktu pentransmisian. Untuk standar paket loss pada IPTV berdasarkan THIPON, *packet loss* dalam kategori sangat bagus adalah 0%, *packet loss* untuk kategori bagus adalah 0% - 3%, *packet loss* untuk kategori sedang adalah 3% - 15%, dan *packet loss* untuk kategori jelek adalah 15% - 25%. Untuk kategori *packet loss* ini untuk semua ukuran paket yang ingin dikirim.

Umumnya perangkat jaringan memiliki *buffer* untuk menampung data yang diterima. Jika terjadi tabrakan yang cukup banyak, *buffer* akan penuh, dan data baru tidak dapat diterima. Paket yang hilang ini harus ditransmisi ulang, yang akan membutuhkan waktu tambahan.

Probabilitas *packet loss* penerapan aplikasi IPTV pada jaringan LTE, ditentukan berdasarkan pada probabilitas *packet loss* pada jaringan LTE dan *server*. Probabilitas *packet loss* total seperti pada persamaan (Pritchard, 1993) :

$$P_{\text{IPTV}} = 1 - [(1 - P_{\text{LTE}})(1 - P_{\text{server}})] \tag{2.20}$$

dimana:

P_{IPTV} = probabilitas *packet loss* IPTV

P_{LTE} = probabilitas *packet loss* pada jaringan LTE

P_{server} = probabilitas *packet loss* pada *server*

Prosentase *packet loss* ditentukan dengan persamaan (Wijaya Hendra A, 2005 : 34) :

$$packet\ loss(\%) = \frac{N_{packet\ loss}}{N_{paket} + N_{packet\ loss}} \times 100\% \quad (2.21)$$

dengan :

- $N_{packet\ loss}$ = jumlah paket yang hilang
- N_{paket} = jumlah paket yang diterima dengan benar

2.4.4.1 Probabilitas *Packet Loss* pada *Server*

Probabilitas *packet loss* pada *server* adalah probabilitas *packet loss* yang terjadi didalam *server* yang terhubung dengan protokol RTP, UDP dan IP. Karena protokol ini tidak dapat menjamin bahwa paket yang dikirim dapat tersampaikan semua. Probabilitas paket *loss* pada *server* dapat dihitung dengan persamaan:

$$\rho_{server} = P_{va\ IPTV} \times \rho_b \quad (2.22)$$

dimana:

ρ_{server} = probabilitas *packet loss* pada *server*

ρ_b = BER (10^{-8})

$P_{va\ IPTV}$ = panjang paket data IPTV (bit)

2.4.4.2 Probabilitas *Packet Loss* pada Jaringan LTE

Probabilitas *packet loss* pada jaringan LTE dengan mode TDD adalah probabilitas *packet loss* yang terjadi pada jaringan LTE. Probabilitas *packet loss* pada jaringan LTE ini dipengaruhi oleh probabilitas bit salah yang terjadi pada jaringan LTE. . Probabilitas *packet loss* pada jaringan LTE sesuai dengan persamaan:

$$\rho_{LTE} = P_{va\ IPTV} \times \rho_{bLTE} \quad (2.23)$$

dengan :

ρ_{LTE} = probabilitas *packet loss* pada jaringan LTE

ρ_{bLTE} = BER LTE pada saat transmisi

$P_{va\ IPTV}$ = panjang paket IPTV yang ditransmisikan pada jaringan LTE (byte)

2.4.5 Kapasitas Kanal

Kapasitas kanal adalah kemampuan kanal untuk mentransmisikan berapa banyak data yang dapat dikirim. Kapasitas kanal pada skripsi ini adalah kapasitas kanal pada jaringan LTE yang dimulai dari *transmitter* sampai ke *user*. Sesuai dengan rumus Shanon maka kapasitas kanal dapat dihitung dengan :

$$C = B \log_2(1 + SNR) \quad (2.24)$$



dengan :

C ; Kapasitas kanal (Mbps)

B : *bandwidth* pada jaringan (Hz)

SNR : *Signal to Noise Ratio* sistem

2.4.6 *Delay End to End* IPTV pada jaringan LTE dengan menggunakan mode TDD

Delay end to end adalah total waktu keseluruhan yang dibutuhkan untuk mengirim data dari *server* sampai ke *user*. *Delay end to end* untuk aplikasi IPTV pada jaringan LTE dengan mode TDD terdiri dari *delay codec* dan *delay* jaringan LTE. Jadi *delay end to end* adalah penjumlahan *delay codec* dan *delay* jaringan LTE. Standar *delay end to end* sesuai dengan referensi International Telecommunications Union (ITU) G.1010 untuk aplikasi multimedia yang interaktif seperti aplikasi pada IPTV adalah kurang dari 1s. Jadi *delay end to end* dapat dihitung dengan persamaan :

$$t_{\text{end to end}} = t_{\text{codec}} + t_{\text{LTE}} \quad (2.25)$$

dengan :

$t_{\text{end to end}}$ = *delay end to end* (s)

t_{codec} = *delay codec* IPTV (s)

t_{LTE} = *delay* pada jaringan LTE (s)

2.4.6.1 *Delay Codec*

Delay codec adalah waktu yang dibutuhkan tiap paket data untuk mengubah suatu sinyal input menjadi nilai biner atau bit bit dan mengubah nilai biner atau bit bit menjadi suatu sinyal output seperti yang dikirimkan. *Delay* ini terdiri dari waktu untuk mengakumulasi sample suara dan video ke dalam *frame* suara dan video, waktu untuk mengkompresi paket suara dan video, waktu untuk memuat *frame* suara dan video ke dalam paket.

Pada skripsi ini, aplikasi IPTV mengirimkan dua paket data, paket audio dan video, maka *delay codec* pada skripsi ini adalah penjumlahan *delay codec* audio dan *delay codec* video. *Delay codec* untuk audio sebesar 40 ms, ini sesuai dengan jenis *codec* audio yang digunakan yaitu *codec audio* G.719. *Delay codec* video sebesar 300 ms, ini sesuai dengan jenis *codec* video yang digunakan yaitu *codec video* H.264/AVC. Sesuai dengan persamaan 2.26 *delay codec* untuk aplikasi IPTV adalah :

$$t_{\text{codec}} = t_{\text{audio}} + t_{\text{video}} \quad (2.26)$$

dengan:

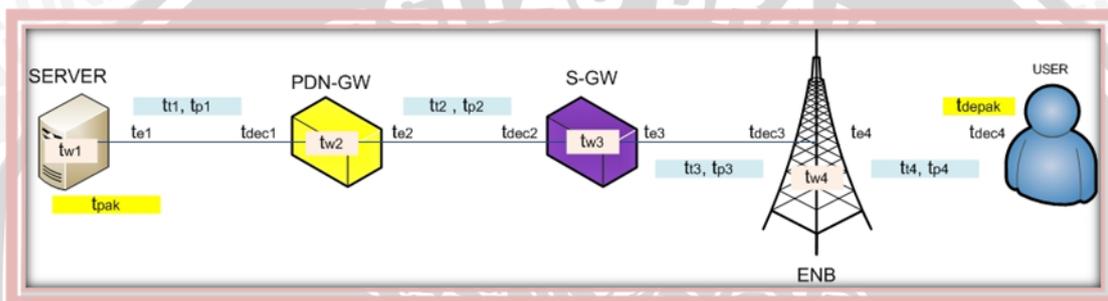
t_{codec} = *delay codec* IPTV (s)

$t_{\text{audio}} = \text{delay codec audio (s)}$

$t_{\text{video}} = \text{delay codec video (s)}$

2.4.6.2 Delay pada Jaringan LTE

Delay jaringan LTE adalah waktu yang dibutuhkan paket data untuk melewati jaringan LTE. Delay pada jaringan LTE terjadi dari server ke PDN – GW ke S- GW ke eNB sampai ke user. Delay pada jaringan LTE terdiri dari delay enkapsulasi, delay dekapsulasi, delay transmisi, delay propagasi, delay antrian, delay depaketisasi dan delay paketisasi. Jadi delay pada jaringan LTE adalah penjumlahan dari delay enkapsulasi, delay dekapsulasi, delay transmisi, delay propagasi, delay antrian, delay depaketisasi dan delay paketisasi.



Gambar 2.19 Delay pada Jaringan LTE

(Sumber: Hasil Perancangan)

Delay pada jaringan LTE dengan mode TDD dapat dihitung dengan :

$$t_{\text{LTE}} = t_{\text{enkapsulasi}} + t_{\text{dekapsulasi}} + t_t + t_p + t_w + t_{\text{paketisasi}} + t_{\text{depaketisasi}} \quad (2.27)$$

dengan :

t_{LTE} = delay end-to-end pada jaringan LTE (s)

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

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

t_t = delay transmisi (s)

t_p = delay propagasi (s)

t_w = delay antrian (s)

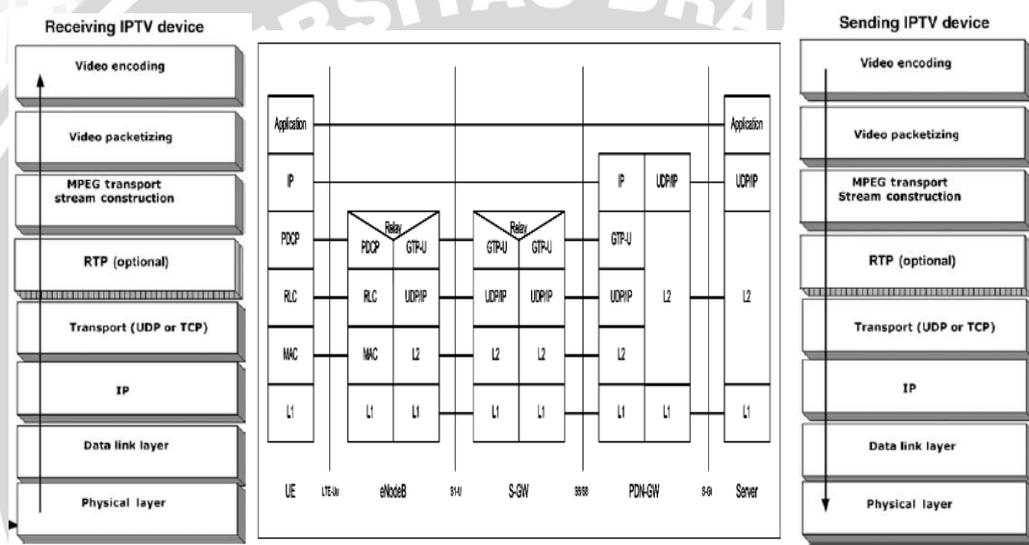
$t_{\text{paketisasi}}$ = delay paketisasi (s)

$t_{\text{depaketisasi}}$ = delay depaketisasi (s)

a. Delay Proses

Delay proses merupakan waktu yang dibutuhkan untuk memproses paket data dan untuk menentukan kemana data tersebut akan diteruskan. Delay proses pada jaringan 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. Dibawah ini adalah gambar proses enkapsulasi dan dekapsulasi pada layer yang terdapat pada jaringan LTE dengan mode TDD.



Gambar 2.20 Blok Proses Enkapsulasi dan Dekapsulasi

(Sumber : Gerard O’Driscoll, 2007)

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

$$t_{enkapsulasi} = \frac{W_{frame} - P}{c} \times 8 \quad (2.28)$$

$$t_{dekapsulasi} = \frac{W_{frame} - P}{c} \times 8 \quad (2.29)$$

dengan:

$t_{enkapsulasi}$ = delay enkapsulasi (s)

$t_{dekapsulasi}$ = delay dekapsulasi (s)

W_{frame} = panjang data (byte)

P = panjang paket data di node (byte)

C = kecepatan pemrosesan data (bps)

Delay enkapsulasi terjadi pada *server* , PDN-GW, S-GW, dan *eNodeB*. Sedangkan *delay* dekapsulasi terjadi di PDN-GW, S-GW, dan *eNodeB*, dan UE.

- *Server*

Pada *server* terjadi proses enkapsulasi sebelum paket data ditransmisikan ke *PDN - GW*. Pada skripsi ini, standard protokol yang digunakan adalah Ethernet dengan MTU (*Maximum Transfer Unit*) *Ethernet* sebesar 1500 *byte* (RFC 895). Oleh karena itu, besar tiap paket data IPTV tidak boleh melebihi MTU sebesar 1500 *byte*.

Apabila panjang paket data IPTV melebihi *Maximum Transfer Unit* (MTU) *Ethernet* sebesar 1500 *byte* (RFC 895), maka paket data IPTV akan mengalami fragmentasi menjadi beberapa buah frame baru. Jika paket data IPTV tidak melebihi MTU *Ethernet*, maka paket data IPTV tidak akan mengalami fragmentasi. Sesuai dengan persamaan berikut :

$$N_{frame} = \frac{P_{va IPTV}}{MTU} \quad (2.30)$$

dengan :

$P_{va IPTV}$ = panjang paket data IPTV (*byte*)

MTU = *Maximum Transfer Unit* (MTU) *Ethernet* sebesar 1500 *byte*

Paket data IPTV diberi penambahan *header* UDP/IP dengan masing masing panjang *header* UDP adalah 8 *byte* (64 *bit*) dan panjang *header* IP 40 *byte* (320 *bit*). Selain itu, karena jaringan LTE adalah sebuah jaringan yang sudah menggunakan IP protokolnya, maka akan digunakan protokol ethernet sebagai media transmisi jaringan IP dan digunakan CRC (*Cyclic Redundancy Check*) untuk mendeteksi kerusakan data dalam proses transmisi dan digunakan untuk memverifikasi integritas data file lain. Selain menambahkan *header* UDP dan IP, akan ditambahkan juga *header ethernet* dan CRC dalam setiap titik *server*, PDN – GW , S – GW pada jaringan LTE karena titik ini terhubung langsung oleh jaringan internet. Panjang *header ethernet* adalah 14 *byte* dan panjang *header* CRC adalah 4 *byte*. Maka, panjang data IPTV pada *server* yang akan dikirimkan ke *PDN – GW* sesuai dengan persamaan 2.31 adalah :

$$W_{server} = P_{va IPTV} + [N_{frame} \times (H_{IP} + H_{UDP} + H_{ethernet} + CRC)] \quad (2.31)$$

dengan:

W_{server} = panjang data pada *server* (*byte*)

$P_{va IPTV}$ = panjang paket data IPTV (*byte*)

H_{IP} = panjang *header* IP (40 *byte*)

H_{UDP} = panjang *header* UDP (8 *byte*)

H_{ethernet} = panjang *header ethernet* (14 byte)

CRC = panjang header CRC (4 byte)

Pada *server* digunakan standar *interface Gigabit Ethernet* dengan kecepatan 1 Gbps. Karena paket data IPTV yang cukup besar, dibutuhkan kecepatan pemrosesan data yang cepat, agar *delay* yang dihasilkan sangat kecil. Sehingga *delay* enkapsulasi pada *server* didapatkan dengan persamaan berikut:

$$t_{\text{enkapsulasi1}} = \frac{W_{\text{server}} - P_{\text{vaIPTV}} \times 8}{C_{\text{server}}} \quad (2.32)$$

dengan:

$t_{\text{enkapsulasi1}}$ = *delay* enkapsulasi pada *server* (s)

W_{server} = panjang data pada *server* (byte)

P_{vaIPTV} = panjang paket data IPTV (byte)

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

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

Setelah dari *server* paket data akan menuju ke PDN –GW. Proses yang dilakukan paket data di PDN – GW yang pertama adalah melakukan dekapsulasi data, yaitu melepas *header header* dari *server*. Sehingga panjang data IPTV pada PDN – GW sesuai dengan persamaan 2.33 adalah

$$W_{\text{PDN-GW}} = W_{\text{server}} - [N_{\text{frames}} \times (H_{\text{IP}} + H_{\text{UDP}} + H_{\text{ethernet}} + \text{CRC})] \quad (2.33)$$

dengan :

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

W_{server} = panjang total data pada *server* (byte)

H_{IP} = panjang *header IP* (40 byte)

H_{UDP} = panjang *header UDP* (8 byte)

H_{ethernet} = panjang *header ethernet* (14 byte)

CRC = panjang header CRC (4 byte)

Pada skripsi ini PDN – GW menggunakan standar *interface Gigabit Ethernet* dengan kecepatan 1 Gbps. Maka besarnya *delay* dekapsulasi pada PDN-GW didapatkan sesuai dengan persamaan:

$$t_{\text{dekapsulasi1}} = \frac{W_{\text{server}} - W_{\text{PDN-GW}} \times 8}{C_{\text{PDN-GW}}} \quad (2.34)$$

dengan:

$t_{\text{dekapsulasi2}}$ = *delay* dekapsulasi pada PDN - GW (s)

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

W_{server} = panjang total data pada *server* (byte)

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

Apabila panjang paket data IPTV pada PDN – GW melebihi *Maximum Transfer Unit* (MTU) *Ethernet* sebesar 1500 *byte* (RFC 895), maka paket data IPTV akan mengalami fragmentasi menjadi beberapa buah *frame* baru. Jika paket data IPTV tidak melebihi MTU *Ethernet*, maka paket data IPTV tidak akan mengalami fragmentasi. Sesuai dengan persamaan berikut :

$$N_{frame} = \frac{W_{PDN-GW}}{MTU} \quad (2.35)$$

Kemudian, sebelum paket data aplikasi IPTV ditransmisikan ke S-GW, dilakukan proses enkapsulasi terlebih dahulu, paket data IPTV dienkapsulasi dengan penambahan *header GTP* (*GPRS Tunneling Protocol*), *UDP* (*User Datagram Protocol*), dan *IP* (*Internet Protocol*) sesuai dengan persamaan berikut:

$$W_{2PDN-GW} = W_{PDN-GW} + [N_{frame} \times (H_{IP} + H_{UDP} + H_{GTP} + H_{ethernet} + CRC)] \quad (2.36)$$

dengan :

$W_{2PDN-GW}$ = panjang total data pada PDN-GW (byte)

W_{PDN-GW} = panjang data pada PDN-GW(byte)

H_{IP} = panjang *header IP* (40 byte)

H_{UDP} = panjang *header UDP* (8 byte)

H_{GTP} = panjang *header GTP* (8 byte)

$H_{ethernet}$ = panjang *header ethernet* (14 byte)

CRC = panjang *header CRC* (4 byte)

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

$$t_{enkapsulasi2} = \frac{W_{2PDN-GW} - W_{PDN-GW}}{C_{PDN-GW}} \times 8 \quad (2.37)$$

dengan :

$t_{enkapsulasi2}$ = *delay* enkapsulasi pada PDN - GW (s)

$W_{2PDN-GW}$ = panjang total data pada PDN-GW (byte)

W_{PDN-GW} = panjang data pada PDN-GW(byte)

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

• *Serving Gateway* (S-GW)

Setelah dari PDN – GW paket data akan menuju ke S –GW. Proses yang dilakukan paket data di S – GW yang pertama adalah melakukan dekapsulasi data, yaitu melepas

header header dari PDN –GW. Sehingga panjang data IPTV pada S – GW sesuai dengan persamaan 2.38 adalah

$$W_{S-GW} = W_{2PDN-GW} - [N_{frame} \times (H_{IP} + H_{UDP} + H_{GTP} + H_{ethernet} + CRC)] \quad (2.38)$$

dengan :

W_{S-GW} = panjang data pada S-GW(byte)

$W_{2PDN-GW}$ = panjang data total pada PDN-GW(byte)

H_{IP} = panjang header IP (40 byte)

H_{UDP} = panjang header UDP (8 byte)

$H_{ethernet}$ = panjang header ethernet (14 byte)

H_{GTP} = panjang header GTP (8 byte)

CRC = panjang header CRC (4 byte)

Pada skripsi ini S – GW menggunakan standar interface Gigabit Ethernet dengan kecepatan 1 Gbps. Maka besarnya delay dekapsulasi pada S-GW didapatkan sesuai dengan persamaan:

$$t_{dekapsulasi2} = \frac{W_{2PDN-GW} - W_{S-GW}}{C_{S-GW}} \times 8 \quad (2.39)$$

dengan :

$t_{dekapsulasi2}$ = delay dekapsulasi pada S - GW (s)

W_{S-GW} = panjang data pada S-GW(byte)

$W_{2PDN-GW}$ = panjang data total pada PDN-GW(byte)

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

Apabila panjang paket data IPTV pada S – GW melebihi Maximum Transfer Unit (MTU) Ethernet sebesar 1500 byte (RFC 895), maka paket data IPTV akan mengalami fragmentasi menjadi beberapa buah frame baru. Jika paket data IPTV tidak melebihi MTU Ethernet, maka paket data IPTV tidak akan mengalami fragmentasi. Sesuai dengan persamaan berikut :

$$N_{frame} = \frac{W_{S-GW}}{MTU} \quad (2.40)$$

Kemudian, sebelum paket data IPTV ditransmisikan ke eNodeB, dilakukan proses enkapsulasi terlebih dahulu, paket data IPTV dienkapsulasi dengan penambahan header GTP (GPRS Tunneling Protocol), UDP (User Datagram Protocol), dan IP (Internet Protocol) sesuai dengan persamaan berikut:

$$W_{2S-GW} = W_{S-GW} + [N_{frame} \times (H_{IP} + H_{UDP} + H_{GTP} + H_{ethernet} + CRC)] \quad (2.41)$$

dengan :

W_{2S-GW} = panjang data total pada S-GW (byte)

W_{S-GW} = panjang data pada S-GW(byte)

H_{IP} = panjang *header* IP (40 byte)

H_{UDP} = panjang *header* UDP (8 byte)

H_{GTP} = panjang *header* GTP (8 byte)

$H_{ethernet}$ = panjang *header ethernet* (14 byte)

CRC = panjang header CRC (4 byte)

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

$$t_{enkapsulasi} = \frac{W_{2S-GW} - W_{S-GW}}{C_{S-GW}} \times 8 \quad (2.42)$$

dengan :

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

W_{2S-GW} = panjang data total pada S-GW (byte)

W_{S-GW} = panjang data pada S-GW(byte)

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

- *Evolved Node B* (ENodeB)

Setelah dari S- GW paket data akan menuju ke eNodeB. Proses yang dilakukan paket data di eNodeB yang pertama adalah melakukan dekapsulasi data, yaitu melepas *header header* dari S- GW. Sehingga panjang data IPTV pada ENodeB sesuai dengan persamaan 2.43 adalah

$$W_{eNB} = W_{2S-GW} - [N_{frame} \times (H_{IP} + H_{UDP} + H_{GTP} + H_{ethernet} + CRC)] \quad (2.43)$$

dengan :

W_{eNB} = panjang data pada eNB(byte)

W_{2S-GW} = panjang data total pada S-GW (byte)

H_{IP} = panjang *header* IP (40 byte)

H_{UDP} = panjang *header* UDP (8 byte)

$H_{ethernet}$ = panjang *header ethernet* (14 byte)

H_{GTP} = panjang *header* GTP (8 byte)

CRC = panjang header CRC (4 byte)

Maka besarnya *delay* dekapsulasi didapatkan dengan menggunakan persamaan:

$$t_{dekapsulasi} = \frac{W_{2S-GW} - W_{eNB}}{C_{eNB}} \times 8 \quad (2.44)$$

dengan :

- $t_{dekapsulasi}$ = delay dekapsulasi pada eNB (s)
 W_{2S-GW} = panjang data total pada S-GW (byte)
 W_{eNB} = panjang data pada eNB(byte)
 C_{eNodeB} =kapasitas kanal di eNodeB (bps)

Apabila panjang paket data IPTV pada eNodeB melebihi *Maximum Transfer Unit* (MTU) *Ethernet* sebesar 1500 byte (RFC 895), maka paket data IPTV akan mengalami fragmentasi menjadi beberapa buah *frame* baru. Jika paket data IPTV tidak melebihi MTU *Ethernet*, maka paket data IPTV tidak akan mengalami fragmentasi. Sesuai dengan persamaan berikut :

$$N_{frame} = \frac{W_{eNB}}{MTU} \quad (2.45)$$

Kemudian, sebelum paket data IPTV ditransmisikan ke UE, dilakukan proses enkapsulasi terlebih dahulu, paket data IPTV dienkapsulasi dengan penambahan *header* PDCP (*Packet Data Convergence Protocol*), RLC (*Radio Link Control*), dan *Medium Access Control* (MAC) .

Pada saat memasuki PDCP (*Packet Data Convergence Protocol*), maka langsung dienkapsulasi dengan menambahkan *header* PDCP sesuai dengan persamaan :

$$W_{1eNB} = W_{eNB} + (N_{frame} \times H_{PDCP}) \quad (2.46)$$

dengan :

- W_{1eNB} = panjang data total 1 pada eNB(byte)
 W_{eNB} = panjang data pada eNB(byte)
 H_{PDCP} = panjang *header* PDCP (2 byte)

Pada saat memasuki RLC, paket data disegmentasi menjadi paket sebesar 40 byte. Jumlah *frame* RLC diketahui sesuai dengan persamaan berikut :

$$N_{RLC} = \frac{W_{1eNB}}{40 \text{ byte}} \quad (2.47)$$

dengan :

- W_{1eNB} = panjang data total 1 pada eNB(byte)
 $N_{frame RLC}$ = jumlah frame pada RLC

Setiap frame RLC diberi *header* RLC sebesar 2 byte sehingga panjang data *frame* RLC setelah penambahan *header* sesuai dengan persamaan :

$$W_{2eNB} = W_{1eNB} + (N_{frame RLC} \times H_{RLC}) \quad (2.48)$$

dengan :

- W_{2eNB} = panjang data total 2 pada eNB(byte)
- W_{1eNB} = panjang data total 1 pada eNB(byte)
- $N_{frame\ RLC}$ = jumlah frame pada RLC
- H_{RLC} =panjang *header* RLC (2 byte)

Pada saat memasuki *layer* MAC, paket data disegmentasi menjadi paket sebesar 42 byte. Jumlah *frame* MAC diketahui sesuai dengan persamaan berikut :

$$N_{frame\ MAC} = \frac{W_{2eNB}}{42\ byte} \quad (2.49)$$

dengan :

- W_{2eNB} = panjang data total 2 pada eNB(byte)
- $N_{frame\ MAC}$ = jumlah frame pada MAC

Setiap frame MAC diberi *header* MAC sebesar 3 byte. Panjang paket data IPTV pada ENodeB yang siap ditransmisikan menuju UE merupakan panjang data *frame* MAC total yaitu :

$$W_{3\ eNB} = W_{2eNB} + (N_{frame\ MAC} \times H_{MAC}) \quad (2.50)$$

dengan :

- W_{3eNB} = panjang data total 3 pada eNB(byte)
- W_{2eNB} = panjang data total 2 pada eNB(byte)
- $N_{frame\ MAC}$ = jumlah frame pada MAC
- H_{MAC} =panjang *header* MAC (3 byte)

Sehingga besar *delay* enkapsulasi yang terjadi pada ENodeB didapatkan dengan menggunakan persamaan:

$$t_{enkapsulasi4} = \frac{W_{3\ eNB} - W_{eNB}}{C_{eNB}} \times 8 \quad (2.51)$$

dengan :

- $t_{enkapsulasi4}$ = *delay* enkapsulasi pada eNB (s)
- W_{3eNB} = panjang data total 3 pada eNB(byte)
- W_{eNB} = panjang data pada eNB(byte)
- C_{eNodeB} =kapasitas kanal di eNodeB (bps)

- *User Equipment* (UE)

Setelah dari ENodeB paket data akan menuju ke UE. Proses yang dilakukan paket data di UE adalah hanya melakukan dekapsulasi data, yaitu melepas *header header* dari ENodeB. Sehingga panjang data IPTV pada UE sesuai dengan persamaan 2.52 adalah

$$W_{UE} = W_{3eNB} - (N_{frame\ RLC} \times H_{RLC}) - (N_{frame\ MAC} \times H_{MAC}) - (N_{frame} \times H_{PDCP}) \quad (2.52)$$

dengan :

- W_{UE} = panjang data pada UE(byte)
- W_{3eNB} = panjang data total 3 pada eNB(byte)
- $N_{frame\ RLC}$ = jumlah frame pada RLC
- H_{RLC} = panjang *header* RLC (2 byte)
- $N_{frame\ MAC}$ = jumlah frame pada MAC
- H_{MAC} = panjang *header* MAC (3 byte)
- H_{PDCP} = panjang *header* PDCP (2 byte)

Maka besarnya *delay* dekapsulasi yang terjadi di UE didapatkan dengan persamaan:

$$t_{dekapsulasi4} = \frac{W_{3eNB} - W_{UE} \times 8}{C_{UE}} \quad (2.53)$$

dengan :

- $t_{dekapsulasi4}$ = *delay* dekapsulasi pada UE (s)
- W_{UE} = panjang data pada UE(byte)
- W_{3eNB} = panjang data total 3 pada eNB(byte)
- C_{UE} = kapasitas kanal pada UE (bps)

Jadi *delay* proses total = $t_{enkapsulasi1} + t_{dekapsulasi1} =$

$$t_{enkapsulasi1} + t_{dekapsulasi1} + t_{enkapsulasi2} + t_{dekapsulasi2} + t_{enkapsulasi3} + t_{dekapsulasi3} + t_{enkapsulasi4} + t_{dekapsulasi4}$$

b. *Delay* Transmisi

Delay transmisi adalah waktu yang dibutuhkan oleh sebuah informasi (bit, byte atau paket) untuk meletakkan keseluruhan bit informasi pada media transmisi. Pada *delay* ini akan dipengaruhi oleh panjang paket data yang akan dikirim dan kecepatan jenis media transmisi yang digunakan. Pada skripsi ini, panjang paket data IPTV yang akan dikirim terdiri dari paket audio dan video. *Delay* transmisi terjadi ketika paket data akan diletakkan 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*.

Delay transmisi dapat ditentukan dengan persamaan (Mischa Schwartz, 1987 : 132):

$$t_t = \frac{W_{data}}{c} \quad (2.54)$$

- *Delay Transmisi dari Server ke PDN – GW*

Setelah dari *server* paket data akan diteruskan ke PDN – GW, sebelum diteruskan ke PDN – GW, dibutuhkan waktu untuk meletakkan paket data ke media transmisi dan berjalan menuju PDN – GW, maka *delay* ini akan dipengaruhi oleh jenis media transmisi yang digunakan untuk menghubungkan server ke PDN – GW, karena jenis media transmisi ini akan menentukan kecepatan media itu untuk dapat mengirimkan paket data, dan jenis media transmisi yang digunakan dari *server* ke PDN – GW adalah kabel UTP kategori 5e dengan kecepatan transmisi maksimal 1Gbps. Karena paket data IPTV yang cukup besar, dibutuhkan kecepatan transmisi yang cepat, agar *delay* yang dihasilkan sangat kecil. Dan panjang paket data pada *server* adalah panjang paket data IPTV ditambah dengan penambahan *header* pada server .

Maka, *delay* transmisi pada *server* ke PDN – GW sesuai persamaan 2.55 adalah

$$t_{t1} = \frac{W_{data}}{c} \quad (2.55)$$

dengan :

t_{t1} = *delay* transmisi dari *server* ke PDN - GW (s)

W_{data} = panjang *data* pada *server* (byte)

C = kecepatan transmisi kanal (bps)

- *Delay Transmisi PDN – GW ke S – GW*

Setelah dari PDN – GW paket data akan diteruskan ke S – GW, sebelum diteruskan ke S– GW, dibutuhkan waktu untuk meletakkan paket data ke media transmisi dan berjalan menuju S – GW, maka *delay* ini akan dipengaruhi oleh jenis media transmisi yang digunakan untuk menghubungkan PDN – GW ke S – GW, karena jenis media transmisi ini akan menentukan kecepatan media itu untuk dapat mengirimkan paket data, dan jenis media transmisi yang digunakan dari PDN – GW ke S – GW adalah kabel UTP kategori 5e dengan kecepatan transmisi maksimal 1Gbps. Dan panjang paket data pada PDN – GW adalah panjang paket data IPTV ditambah dengan penambahan *header* pada PDN – GW .

Maka, *delay* transmisi pada PDN – GW ke S – GW sesuai persamaan 2.56 adalah

$$t_{t2} = \frac{W_{data}}{c} \quad (2.56)$$

dengan :

t_{t2} = *delay* transmisi dari PDN – GW ke S – GW (s)

W_{data} = panjang *data* pada *PDN - GW* (byte)

C = kecepatan transmisi kanal (bps)

- *Delay* Transmisi *S – GW* ke *ENodeB*

Setelah dari *S– GW* paket data akan diteruskan ke *ENodeB*, sebelum diteruskan ke *ENodeB* , dibutuhkan waktu untuk meletakkan paket data ke media transmisi dan berjalan menuju *ENodeB*, maka *delay* ini akan dipengaruhi oleh jenis media transmisi yang digunakan untuk menghubungkan *S – GW* ke *ENodeB*, karena jenis media transmisi ini akan menentukan kecepatan media itu untuk dapat mengirimkan paket data, dan jenis media transmisi yang digunakan dari *S – GW* ke *ENodeB* adalah kabel fiber jenis monomode fiber dengan kecepatan transmisi maksimal 1Gbps. Untuk media transmisi pada *S – GW* ke *ENodeB* memakai jenis kabel fiber karena jarak transmisi antara *S – GW* ke *ENodeB* cukup jauh, yaitu 1000 m. Dan panjang paket data pada *S– GW* adalah panjang paket data *IPTV* ditambah dengan penambahan *header* pada *S– GW* .

Maka, *delay* transmisi pada *S – GW* ke *ENodeB* sesuai persamaan 2.57 adalah

$$t_{t3} = \frac{W_{data}}{C} \quad (2.57)$$

dengan :

t_{t3} = *delay* transmisi dari *S – GW* ke *ENodeB* (s)

W_{data} = panjang *data* pada *S - GW* (byte)

C = kecepatan transmisi kanal (bps)

- *Delay* Transmisi *ENodeB* ke *UE*

Setelah dari *ENodeB* paket data akan diteruskan ke *UE*, sebelum diteruskan ke *UE* , dibutuhkan waktu untuk meletakkan paket data ke media transmisi dan berjalan menuju *UE*. Pada skripsi ini *eNodeB* mengirimkan paket data ke *UE* melalui media transmisi udara. Dan panjang paket data pada *ENodeB* adalah panjang paket data *IPTV* ditambah dengan penambahan *header* pada *ENodeB*. Selain itu, karena pada jaringan *LTE* ini menggunakan mode *TDD* yang mempunyai prinsip kerja yaitu *TDD* dapat bekerja pada frekuensi *uplink* dan *downlink* yang sama tetapi dialokasikan ke dalam slot-slot waktu yang berbeda dan saling dipisahkan oleh selang waktu (*guard time*), maka dari itu paket data yang akan diletakkan ke media transmisi dan akan dikirim ke user harus dibagi sesuai dengan time slot. Jumlah slot ditentukan oleh *chip rate* *TDD-LTE* yaitu sebesar 3,84 Mcps. *Chip rate* sebesar 3,84 Mcps selanjutnya dibagi menjadi

radio *frame* 10 ms. Satu slot terdiri dari 2560 *chip* dan banyaknya slot dapat dihitung menggunakan persamaan $n = \frac{(cr \times rf)}{2560 \text{ chip}}$,

dengan:

n = banyaknya slot

cr = *chip rate*

rf = *radio frame* (Harri Holma dan Otto Lehtien, 2009 : 353).

Maka, *delay* transmisi pada ENodeB ke UE sesuai persamaan 2.58 adalah

$$t_{t4} = \frac{W_{data}}{n \times C} \times 8 \quad (2.58)$$

dengan :

t_{t4} = *delay* transmisi dari ENodeB ke UE (s)

W_{data} = panjang *data* pada ENodeB byte)

C = kecepatan transmisi kanal (bps)

Sehingga *delay* transmisi total ditentukan dengan persamaan:

$$t_{t \text{ total}} = t_{t1} + t_{t2} + t_{t3} + t_{t4} \quad (2.59)$$

dengan :

$t_{t \text{ total}}$ = *delay* transmisi total (s)

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 - eNB (s)

t_{t4} = *delay* transmisi pada eNB - UE (s)

c. *Delay Propagasi*

Delay propagasi adalah waktu yang dibutuhkan untuk merambatkan paket data melalui media transmisi dari *server* sampai UE. Dengan panjang paket data IPTV yang dikirim terdiri dari paket audio dan video. *Delay* ini dipengaruhi oleh panjang jarak antar *node* dan kecepatan media gelombang propagasi. *Delay* ini terjadi ketika paket data merambat melalui media transmisi dari server ke PDN – GW, PDN – GW ke S – GW, S – GW ke ENodeB, dan ENodeB ke UE. *Delay* propagasi dapat ditulis dengan :

$$t_p = \frac{d_{max}}{v} \quad (2.60)$$

✓ *Delay* Propagasi antar server ke PDN – GW

Delay propagasi dari *server* ke PDN – GW dipengaruhi oleh panjang jarak yang ditetapkan dari *server* ke PDN – GW yaitu 10 m, dan kecepatan media gelombang

propagasi 3×10^8 m/s. Maka sesuai dengan persamaan 2.61 *delay* propagasi dari *server* ke PDN – GW adalah

$$t_{p1} = \frac{d_{max}}{v} \quad (2.61)$$

dengan :

t_{p1} = *delay* propagasi dari *server* ke PDN – GW (s)

d_{max} = panjang *jarak* dari *server* ke PDN – GW (m)

v = kecepatan media gelombang propagasi 3×10^8 m/s

✓ *Delay* Propagasi antar PDN – GW ke S – GW

Delay propagasi dari PDN – GW ke S – GW dipengaruhi oleh panjang *jarak* yang ditetapkan dari PDN – GW ke S – GW yaitu 10 m, dan kecepatan media gelombang propagasi 3×10^8 m/s. Maka sesuai dengan persamaan 2.62 *delay* propagasi dari PDN – GW ke S – GW adalah

$$t_{p2} = \frac{d_{max}}{v} \quad (2.62)$$

dengan :

t_{p2} = *delay* propagasi dari PDN – GW ke S – GW (s)

d_{max} = panjang *jarak* dari PDN – GW ke S – GW (m)

v = kecepatan media gelombang propagasi 3×10^8 m/s

✓ *Delay* Propagasi antar S – GW ke ENodeB

Delay propagasi dari S – GW ke ENodeB dipengaruhi oleh panjang *jarak* yang ditetapkan dari S – GW ke ENodeB yaitu 1000 m, dan dan kecepatan media gelombang propagasi 3×10^8 m/s. Maka sesuai dengan persamaan 2.63 *delay* propagasi dari S – GW ke ENodeB adalah

$$t_{p3} = \frac{d_{max}}{v} \quad (2.63)$$

dengan :

t_{p3} = *delay* propagasi dari S – GW ke ENodeB (s)

d_{max} = panjang *jarak* dari S – GW ke ENodeB (m)

v = kecepatan media gelombang propagasi 3×10^8 m/s

✓ *Delay* Propagasi antar ENodeB ke UE

Delay propagasi dari ENodeB ke UE dipengaruhi oleh panjang *jarak* ENodeB ke UE yang berubah yaitu 1000 m – 10000 m, dan dan kecepatan media gelombang

propagasi 3×10^8 m/s. Maka sesuai dengan persamaan 2.64 *delay* propagasi dari ENodeB ke UE pada jarak ENodeB ke UE 1000 m adalah

$$t_{p4} = \frac{d_{max}}{v} \quad (2.64)$$

dengan :

t_{p4} = *delay* propagasi dari ENodeB ke UE (s)

d_{max} = panjang *jarak* dari ENodeB ke UE (m)

v = kecepatan media gelombang propagasi 3×10^8 m/s

Sehingga *delay* propagasi total ditentukan dengan persamaan:

$$t_{p\ total} = t_{p1} + t_{p2} + t_{p3} + t_{p4} \quad (2.65)$$

dengan:

$t_{p\ total}$ = *delay* propagasi total (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 - eNB (s)

t_{p4} = *delay* propagasi pada eNB - UE (s)

d. *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 ini dipengaruhi oleh panjang paket data yang akan dikirim, kecepatan pemrosesan data pada *node* itu, dan faktor utilisasi. Faktor utilisasi mengindikasikan banyaknya *user* yang berada dalam antrian (Dimitri Bertsekas and Robert Gallager, 1987: 129). Pada skripsi ini faktor utilisasi yang digunakan adalah 9/10, yang dapat diartikan bahwa terdapat 9 *user* dalam antrian yang berkapasitas maksimal 10 *user*, karena cara pengirimannya adalah *multicast*. *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 jumlah *server* yang akan melayani pelanggan. *Delay* antrian ini terjadi pada *server*, PDN – GW, S – GW, ENodeB. *Delay* antrian ditentukan dengan persamaan sebagai berikut :

$$t_W = \frac{1}{\mu} + \frac{\lambda_W}{\mu^2(1-\rho)} \quad (2.66)$$

dengan:

t_w : *delay* antrian (s)

μ : kecepatan pelayanan paket (paket/s)

λ_w : kecepatan kedatangan paket (paket/s)

ρ : faktor utilitas

Sedangkan kecepatan pelayanan paket ditentukan dengan persamaan sebagai berikut :

$$\mu = \frac{C}{L} \quad (2.67)$$

dengan :

μ : kecepatan pelayanan paket (paket/s)

C : kapasitas kanal (bps)

L : panjang paket data (bit)

Dan untuk menentukan kecepatan kedatangan paket digunakan persamaan :

$$\lambda_w = \mu \times \rho \quad (2.68)$$

dengan:

λ_w : kecepatan kedatangan paket (paket/s)

μ : kecepatan pelayanan paket (paket/s)

ρ : faktor utilitas

Jadi *delay* antrian pada jaringan LTE dengan mode TDD :

$$t_{w\text{total}} = t_{w1} + t_{w2} + t_{w3} + t_{w4} \quad (2.69)$$

dengan :

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

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

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

t_{w3} = *delay* antrian pada S-GW (s)

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

e. *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 oleh panjang paket data dan kecepatan pemrosesan data. *Delay* paketisasi dinyatakan dengan persamaan sebagai berikut :

$$t_{\text{paketasasi}} = \frac{L_{\text{paket}}}{C} \tag{2.70}$$

dengan :

L_{paket} : besar paket yang dikirimkan (byte)

C : kapasitas kanal transmisi (bps)

Sedangkan untuk *delay* depaketisasi dinyatakan dengan persamaan sebagai berikut :

$$t_{\text{depaketisasi}} = \frac{L_{\text{paket}}}{C} \tag{2.71}$$

dengan:

L_{paket} : besar paket yang diterima (byte)

C : kapasitas kanal transmisi (bps)

2.4.7 Perhitungan *Throughput*

Throughput adalah kecepatan maksimum jaringan saat tidak ada data yang hilang pada pentransmisiannya atau banyaknya data yang diterima dengan benar oleh *user*. Analisis perhitungan *throughput* yang dilakukan dalam skripsi ini adalah hubungan antara *transmitter* hingga ke *user*. *Throughput* dipengaruhi oleh probabilitas *packet loss* , *delay* total transmisi pengiriman paket data melalui media transmisi mulai dari *server* sampai dengan *user* dan konstanta propagasi (α). Konstanta ini dipengaruhi oleh *delay* propagasi, yaitu waktu total yang dibutuhkan paket data untuk merambat dengan gelombang melalui media transmisi dari *server* sampai *user*. *Throughput* dapat dihitung dengan persamaan (Mischa Schwartz, 1987:129)

$$\lambda = \frac{(1-\rho_{IPTV})}{t_t[1+(\alpha-1)\rho_{IPTV}]} \tag{2.72}$$

dengan :

λ : *Throughput* (paket/s)

ρ_{IPTV} : probabilitas *packet loss*

t_t : *delay* transmisi (s)

Simbol α merupakan konstanta propagasi dengan persamaan sebagai berikut :

$$\alpha = \left(3 + \frac{2t_p}{t_t} \right) \tag{2.73}$$

dengan :

α : konstanta propagasi

t_p : *delay* propagasi(s)

t_t : *delay* transmisi (s)



