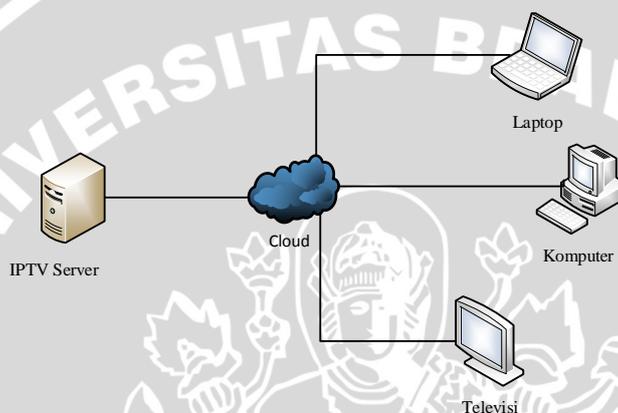


BAB II

TINJAUAN PUSTAKA

2.1 Internet Protocol Television (IPTV)

Internet Protocol Television (IPTV) adalah teknologi yang memanfaatkan *Internet Protocol* (IP) untuk menyediakan layanan gambar bergerak (*video*) beserta suara (*audio*) dalam bentuk paket secara *real-time*. Gambar 2.1 menunjukkan model sistem layanan IPTV secara umum.



Gambar 2.1 Model Sistem Layanan IPTV
(Sumber: Wauters, 2006)

Internet Protocol Television (IPTV) terbagi menjadi dua tipe yakni *IPTV point to point* dan *IPTV multipoint*.

1. *Internet Protocol Television* (IPTV) *point to point* melibatkan dua *client* atau lebih yang terintegrasi melalui *single ip address*.
2. *Internet Protocol Television* (IPTV) *multipoint* melibatkan *client* dan sebuah *server*

2.1.1 Codec Layanan IPTV

a. H.264

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 *MPEG-4 Advanced Video Coding* (AVC). H.264 adalah standar kompresi *video* yang mempunyai kemampuan kompresi lebih baik dari pendahulunya bahkan jika digunakan pada *data rate* dibawa 1 Mbps. Melalui *MPEG-4 AVC*, *video coding* sangatlah efisien, dalam hal ini membuat integrasi grafis yang

dihasilkan komputer dengan berbagai objek yang tersedia sangatlah sederhana dan efisien dalam penggunaan *bandwidth*.

b. AAC

AAC adalah standar *audio* yang digunakan oleh pengkodean PCM. PCM mengirimkan *sample audio* sebagai pulsa dan dikodekan untuk dinyatakan dalam *amplitude* sinyal aslinya. PCM pada umumnya memerlukan *bandwidth* 64 Kbps pada *client*. Spesifikasi *codec audio* dan *video* yang digunakan pada penelitian ditunjukkan pada tabel 2.1.

Tabel 2.1 Spesifikasi *Audio dan Video Codec*

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

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

2.1.2 Protocol Pendukung IPTV

Protocol adalah sebuah aturan yang mengatur komunikasi data. *Protocol* merepresentasikan sebuah persetujuan antara dua perangkat. Tanpa *protocol*, dua perangkat mungkin tersambung namun tidak dapat berkomunikasi, sama halnya seperti seseorang yang berbahasa Prancis tidak dapat dipahami oleh orang yang berbahasa Jepang. (Forouzan, 2007: 5). Ada beberapa *protocol* yang dapat digunakan dalam IPTV, antara lain:

a. Transfer Control Protocol (TCP)

TCP adalah *protocol transport* aliran data yang handal. *Protocol* ini bersifat *connection-oriented*. Sebuah koneksi harus sudah terpasang antara ujung-ujung dari transmisi sebelum keduanya bisa mentransmisikan data. TCP terletak pada *transport layer* dari *protocol* TCP/IP. Walaupun bersifat *connection-oriented*, TCP tidak cocok digunakan untuk trafik multimedia yang interaktif karena TCP tidak memiliki fitur untuk mentransmisikan ulang paket yang hilang/rusak.

b. User Datagram Protocol (UDP)

UDP adalah *protocol* yang lebih sederhana dari dua standar *protocol* pada *layer transport* dari TCP/IP. *Protocol* ini adalah *protocol* proses ke proses yang hanya menambahkan alamat *port*, *error control checksum*, dan panjang informasi data dari *layer* lebih atas. *Protocol* ini bersifat *connection-less*. UDP lebih cocok digunakan untuk trafik multimedia yang interaktif. Namun UDP tetap membutuhkan *protocol* tambahan yakni RTP.

c. *Real-time Transport Protocol (RTP)*

RTP adalah *protocol* yang didesain untuk menangani trafik *real-time* dari *internet*. RTP tidak memiliki mekanisme (*multicast*, *port numbers* dan lain-lain). RTP harus digunakan bersamaan dengan UDP. Kontribusi utama dari RTP adalah *time-stamping*, *sequencing* dan fasilitas *mixing*.

d. *Internet Protocol (IP)*

IP merupakan sebuah *protocol* jaringan yang berada pada jaringan komunikasi yang terkoneksi dengan *Local Area Network (LAN)* maupun *Wide Area Network (WAN)*. IP terdiri dari informasi pengalamatan dan beberapa kontrol yang memungkinkan suatu data agar dapat dirutekan.

2.1.3 *Bandwidth Aplikasi IPTV*

Jumlah *bandwidth* yang diperlukan dalam aplikasi IPTV sangat bergantung pada ukuran *frame*, kecepatan *frame* dan jenis pengkodean yang digunakan. Untuk menghitung besar *payload video* dan *payload audio* pada IPTV ditentukan dengan persamaan 2-1 dan 2-2 (Wes Simpson, 2008):

$$P_v = B_{CODEC-V} \times \text{frame size video} \quad (2-1)$$

$$P_A = B_{CODEC-A} \times \text{frame size audio} \quad (2-2)$$

Keterangan:

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

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

$B_{CODEC-V}$ = Besarnya kecepatan *codec video* (bps)

$B_{CODEC-A}$ = Besarnya kecepatan *codec audio* (bps)

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

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

Jumlah *payload* yang disegmentasi berdasarkan *payload* maksimum dari ketentuan *codec*. Sehingga, jumlah paket data *video* dan *audio* sesuai dengan persamaan 2-3 dan 2-4 (Wes Simpson, 2008).

$$N_v = \frac{P_v}{P_v \text{maksimum}} \quad (2-3)$$

$$N_a = \frac{P_a}{P_{a \text{ maksimum}}} \quad (2-4)$$

Keterangan:

N_v = Jumlah paket *video*

N_a = Jumlah paket *audio*

$P_v \text{ maksimum}$ = *Payload video* maksimum berdasarkan *codec (bit)*

$P_a \text{ maksimum}$ = *Payload audio* maksimum berdasarkan *codec (bit)*

Payload IPTV (video dan audio) masing-masing ditambahkan *header NALU, RTP, UDP dan IPv4*. Besarnya paket *video dan audio* setelah dienkodakan dan ditambah *header* sesuai dengan persamaan 2-5 dan 2-6 (Wes Simpson, 2008).

$$P_v (\text{encoded}) = P_v + N_v \times (H_{NALU} + H_{RTP} + H_{UDP} + H_{IPv4}) \quad (2-5)$$

Dan untuk *audio*,

$$P_a (\text{encoded}) = P_a + N_a \times (H_{NALU} + H_{RTP} + H_{UDP} + H_{IPv4}) \quad (2-6)$$

Keterangan:

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

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

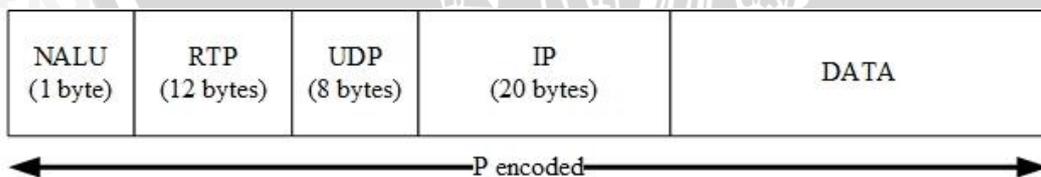
H_{NALU} = *Header NALU (byte)*

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

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

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

Gambar 2.2 menunjukkan format *datagram payload*



Gambar 2.2 Format *Datagram UDP RTP Payload*

Besarnya paket data aplikasi IPTV yang ditransmisikan pada sistem ditentukan dengan persamaan 2-7 (Wes Simpson, 2008).

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

Keterangan:

Piptv size = Panjang *datagram* IPTV (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)

2.2 Virtual Local Area Network (VLAN)

VLAN adalah suatu *broadcast domain* yang dibuat pada sebuah *managable switch* dan memakai sebuah *subset* dari *port* fisik pada *switch*. *Broadcast domain* adalah kumpulan alat jaringan dimana sebuah *broadcast frame* yang dikirim oleh satu alat diterima oleh seluruh alat yang terdapat dalam kumpulan alat jaringan tersebut, suatu LAN fisik dapat disamakan suatu *broadcast domain* (Odom 2004).

Pada dasarnya, VLAN tidak berbeda jauh dengan LAN. Pada LAN, untuk satu peralatan jaringan seperti *switch*, hanya dapat dibuat satu LAN di dalamnya, sedangkan untuk VLAN tidak. VLAN menggunakan suatu jalur *virtual* yang dibuat dari konfigurasi *port-port switch* yang mengizinkan adanya keterhubungan antar *host* yang ada dan terhubung secara logika walaupun tidak terhubung secara langsung dengan menggunakan media kabel.

2.2.1 Kendali Terhadap Penyiaran (*Broadcast Control*)

Penyiaran berada pada setiap *protocol*, tetapi seberapa sering penyiaran ini berlangsung tergantung pada 3 kondisi, yaitu (Jhon, 2004) :

1. Tipe *Protocol*
2. Aplikasi yang berjalan pada jaringan
3. Bagaimana aplikasi-aplikasi ini berjalan pada jaringan.

Beberapa layanan memerlukan *bandwidth* yang kecil dan beberapa lagi memerlukan *bandwidth* yang sangat besar seperti IPTV. Disinilah fungsi dari VLAN untuk membagi *bandwidth* yang ada sesuai dengan layanan yang dijalankan pada suatu jaringan. Jika suatu layanan dijalankan, maka *switch* yang dikonfigurasi dengan VLAN akan menyiarkannya pada pengguna yang menggunakan layanan yang sejenis.

2.2.2 Keamanan

Keamanan menjadi isu yang sangat penting saat ini. Untuk itulah VLAN digunakan untuk mengatasi masalah ini. Administrator jaringan dapat dengan mudah mengawasi dengan sangat teliti pengguna yang akan mengakses layanan tertentu.

2.2.3 Klasifikasi VLAN

Berdasarkan perbedaan kriteria, VLAN dapat diklasifikasikan pada beberapa kategori. Ada beberapa tipe yang bisa digunakan yaitu :

1. *Port Based*

Pada *port-based* VLAN, setiap VLAN diberi nama pada satu atau lebih *port* VLAN yang ada pada *switch*. Pada kondisi awal setiap *port* VLAN dinamakan dengan VLAN ID 1. Jika setiap *user* yang ingin berkomunikasi satu dengan yang lain walaupun dengan IP VLAN yang berbeda jalur harus dialihkan oleh *switch*, bahkan jika setiap *user* tersebut secara fisik pada bagian I/O yang sama. Hal ini menunjukkan setiap VLAN harus dikonfigurasi dengan alamat IP.

Untuk membuat VLAN dengan menggunakan dua *switch*, pengguna perlu menetapkan *port* pada setiap *switch* ke VLAN dan untuk menghubungkan dua *switch* atau lebih (*menggunakan Link Resilient atau Link Aggregation*) pada setiap *switch* per VLAN.

2. *Tagged based*

Tagged VLAN adalah proses menandai ke *Ethernet frame*. *Tag* VLAN memuat nomor identifikasi dari VLAN, yang biasa disebut dengan VLAN ID. Setiap VLAN ditetapkan pada 802.1Q VLAN *tag*. *Port switch* dapat ditambahkan ke VLAN dengan 802.1Q yang telah ditetapkan.

Sebuah VLAN dapat ditandai pada beberapa *port* dan sebaliknya. *Switch* menentukan jalur dari *port* yang ditandai atau yang tidak ditandai secara *real time*. Salah satu keuntungan *tag* VLAN adalah *port* yang digunakan bisa menjadi anggota beberapa VLAN. Hal ini berguna jika menggunakan *server* dari setiap VLAN.

3. *IP Subnet Based*

Header VLAN adalah komunikasi *header* secara *virtual* yang digunakan antara VLAN yang berbeda. Setiap VLAN memiliki satu *header* VLAN. Paket dari sebuah VLAN dapat ditunjukkan pada *layer* jaringan melalui *header* VLAN yang sesuai. Sebagai setiap bentuk alamat *broadcast* dari VLAN, sebuah VLAN dapat menjadi *segmen* jaringan IP tersendiri dan *header* VLAN akan menjadi gerbang yang mengizinkan IP *address* sesuai dengan *layer* tiga yang dituju. (Zeng Xiyang, 2009:2)

2.3 Perangkat Keras Pendukung Jaringan VLAN

Pada penerapannya, teknologi VLAN membutuhkan perangkat yang sama seperti LAN. Dalam pembuatan jaringan VLAN ada beberapa peralatan yang dibutuhkan, seperti

switch, *router*, kabel UTP. Perangkat tersebut masing-masing memiliki fungsi yang berbeda-beda.

2.3.1 Switch

Switch merupakan perangkat keras penghubung di dalam jaringan komputer yang lebih banyak digunakan saat ini dibandingkan *hub*. Hal ini disebabkan karena dengan fungsi yang serupa dengan *hub*, *switch* memiliki dua buah kelebihan utama dibandingkan *hub*. Kelebihan-kelebihan yang dimiliki oleh *switch*, yaitu :

1. *Switch* memiliki kemampuan untuk membaca alamat fisik (*MAC address*) dari setiap komputer yang terhubung ke dalam *switch* bersangkutan. *Switch* menyimpan alamat fisik (*MAC adrees*) dari setiap komputer yang terhubung ke dalam *switch* tersebut beserta dengan nomor *port switch* yang digunakan oleh komputer bersangkutan.
2. *Switch* memiliki kemampuan untuk melakukan *filter* terhadap paket data yang keluar masuk *switch*. Hal ini akan memberikan keamanan paket data (terkait dengan pencarian paket data didalam jaringan komputer). Hal ini terpenting lainnya adalah memberikan kemudahan didalam memberikan informasi mengenai alamat tujuan untuk komputer penerima (*destination address*) serta kemampuan untuk keluar menuju ke komputer tujuan (*outgoing port*)

Gambar 2.3 merupakan perangkat *switch*. *Switch* bekerja di dua buah *layer* pada jaringan komputer, yaitu *data link layer* dan *physical layer*. Pada *data link layer*, terjadi proses pengecekan terhadap alamat fisik jaringan (*MAC address*) untuk otentifikasi alamat fisik komputer yang terhubung pada *switch*, untuk kemudian disesuaikan dengan alamat jaringan pada *network layer* (*IP address*). Pada *physical layer* terjadi proses pengolahan sinyal digital. (Pratama, 2014:487)



Gambar 2.3 Switch
(Sumber : www.cisco.com)

Switch yang digunakan dalam perencanaan ini adalah jenis *switch manageable*. Jenis *switch* ini dapat dikonfigurasi untuk beberapa layanan, seperti *port VPN*, *bandwidth control*, *port trunking*, *port-based VLAN*.

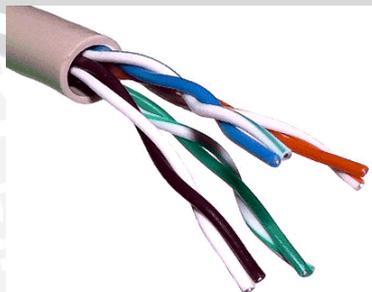
Konfigurasi *port VLAN* dilakukan dengan masuk ke dalam konfigurasi *switch*. Pada *port* tersebut pembagian *VLAN* dilakukan berdasarkan *tagged VLAN*, dimana pengguna dapat menandakan *port* mana saja yang dijadikan *port VLAN*. Sistem kerja *port* tersebut adalah dengan memilih data yang sesuai dengan *port VLAN* yang dituju atau diteruskan.

2.3.2 Kabel UTP

Kabel UTP (*Unshielded Twisted Pair*) adalah kabel jaringan yang paling umum dan paling banyak digunakan di dalam kehidupan sehari-hari untuk jenis komputer bermedia kabel (*wired network*). Kabel jaringan UTP (*Unshielded Twisted Pair*) ini memiliki setidaknya delapan buah kategori, namun yang umum digunakan di dalam kehidupan sehari-hari *Cat5e (Enhanced Category 5)* dan *Cat5 (Category 5)*. Hal ini disebabkan oleh karena kedua kategori dari UTP (*Unshielded Twisted Pair*) memiliki dukungan transmisi paket data di dalam jaringan komputer sebesar 100 Mbps.

Gambar 2.4 merupakan kabel UTP *category 5*. Di dalam merangkai sebuah jaringan komputer memanfaatkan kabel UTP (*Unshielded Twisted Pair*) setidaknya perlu diketahui mengenai dua jenis pengkabelan yang umum digunakan. Kedua jenis tersebut meliputi *Straight Through* dan *Cross Over*.

1. *Straight Through* umum digunakan pada dua buah perangkat yang berbeda di dalam jaringan komputer. Misalkan saja untuk menghubungkan komputer ke perangkat *switch* maupun *hub*.
2. *Cross Over* umum digunakan pada dua buah perangkat yang sama di dalam jaringan komputer. Misalkan untuk menghubungkan dua buah komputer secara langsung (*point to point*) ataupun menghubungkan dua buah *switch* atau *hub* antar jaringan. (Pratama, 2014:526)



Gambar 2.4 Kabel UTP

2.4 Worldwide Interoperability for Microwave Access (WiMAX)

Worldwide Interoperability for Microwave Access (WiMAX) merupakan teknologi akses nirkabel pita lebar (*Broadband Wireless Access*) yang memiliki kecepatan transmisi yang dapat mencapai hingga 75 Mbps serta jarak jangkauan hingga 50 km. Melalui badan *Institute of Electrical and Electronic Engineers (IEEE)* bahwa pengumuman keputusan yang membentuk kelompok kerja 802.16, diupayakan dapat mengembangkan standarisasi bagi teknologi akses nirkabel masa depan dengan memenuhi kriteria sebagai berikut :

1. Memanfaatkan jalur komunikasi nirkabel gelombang mikro
2. Memanfaatkan pita-pita berlisensi
3. Menyediakan layanan jaringan publik
4. Mendukung kemampuan transmisi pita lebar (*broadband*)

Sebuah kelompok industri forum WiMAX dibentuk sebagai salah satu upaya untuk tetap mempromosikan penggunaan standar 802.16 serta mengembangkan spesifikasi-spesifikasi interoperabilitas. Tabel 2.2 menguraikan standar-standar IEEE 802.16 dan ruang lingkup.

Tabel 2.2 Standar-Standar IEEE 802.16

Standar IEEE	Ruang Lingkup
IEEE 802.16-2004	Standar IEEE untuk LAN dan MAN: Antarmuka udara untuk sistem akses nirkabel pita lebar stasioner
IEEE 802.16.2-2004	Praktik rekomendasi IEEE untuk LAN dan MAN: Koeksistensi sistem-sistem akses nirkabel pita lebar stasioner
IEEE 802.16 Conformance (01- 04)	Standar IEEE untuk konfirmasi terhadap 802.16 IEEE (01-03) untuk 10 – 66 GHz dan Bagian (04) di bawah 11 GHz
IEEE 802.16.e-2005	Standar IEEE untuk LAN dan MAN Bagian 16: Lapisan fisik dan MAC pada pita <i>frekuensi</i> berlisensi
IEEE 802.16.f-2005	Standar IEEE untuk LAN dan MAN Bagian 16: <i>Management Information Base</i>
IEEE 802.16.g-2005	Standar IEEE untuk LAN dan MAN Bagian 16: <i>Management Plane Procedures and Services</i>
IEEE 802.16k-2007	Standar IEEE untuk <i>Bridge MAC</i>

(Sumber: William Stallings 2005)

Kemudian *Wireless Area Metropolitan Networks* (WMAN) atau biasa yang lebih populer dikenal WiMAX memiliki standar berdasarkan pada IEEE 802.16 dan juga diterapkan pada *Telecommunications Standards Institute's* (ETSI) yaitu standar pada *High Performance Radio Metropolitan Area Network* (HiperMAN) dengan mengikuti standar yang mendunia. Tabel 2.3 merupakan tabel standar WiMAX 802.16d dan 802.16e.

1. Standar 802.16 dikelompokkan pada rentang transmisi (10 – 66) GHz
2. 802.16d termasuk *frekuensi* rendah yang bekerja pada rentang *frekuensi* (2-11) GHz
3. *Frekuensi-frekuensi* yang umum untuk 802.16d adalah 3.5 GHz dan 5.8 GHz
4. Sedangkan 802.16e adalah 2.3 GHz, 2.5 GHz, dan 3.5 GHz, semuanya fleksibel mengikuti kebijakan dari setiap negara.

Tabel 2.3 Standar WiMAX IEEE 802.16d dan 802.16e

	802.16	802.16d HiperMAN	802.16e
<i>Completed</i>	<i>December 2011</i>	<i>June 2004</i>	<i>2005</i>
<i>Spectrum</i>	<i>(10 – 66) GHz</i>	<i>< 11 GHz</i>	<i>< 6 GHz</i>
<i>Channel</i>	<i>Line-of-sight</i>	<i>Nonline-of-sight</i>	<i>Nonline-of-sight</i>
<i>Conditions</i>	<i>Service only</i>	<i>Service</i>	<i>Service</i>
<i>Bit rate</i>	<i>(32 – 134) Mbps in 28 MHz Channel Bandwidth</i>	<i>Up to 75 Mbps in 20 MHz Channel Bandwidth</i>	<i>Up to 15 Mbps in 5 MHz Channel Bandwidth</i>
<i>Modulation</i>	<i>QPSK, 16 QAM and 64 QAM</i>	<i>OFDM 256 FFT, QPSK, 16 QAM, 64 QAM</i>	<i>Scalable OFDMA, 128- 2.048 FFT/BPSK, QPSK, 16 QAM, 64 QAM</i>
<i>Mobility</i>	<i>Fixed</i>	<i>Fixed</i>	<i>Nomadic/mobile</i>
<i>Channel Bandwidth</i>	<i>20 MHz, 25 MHz and 28 MHz</i>	<i>(1.75 – 20) MHz</i>	<i>(1.75 – 20) MHz</i>

(Sumber: Zerihun Abate, WiMAX RF System Engineering 2009)

2.4.1 Topologi Jaringan WiMAX

Teknologi WiMAX mempunyai beberapa topologi jaringan yaitu jaringan *point to point* dan jaringan *point to multipoint*.

a. *Point to Point*

Point to Point adalah jaringan yang menghubungkan antara dua terminal. Antara sisi pemancar dan sisi penerima terdapat satu perangkat pemancar dan satu perangkat penerima.

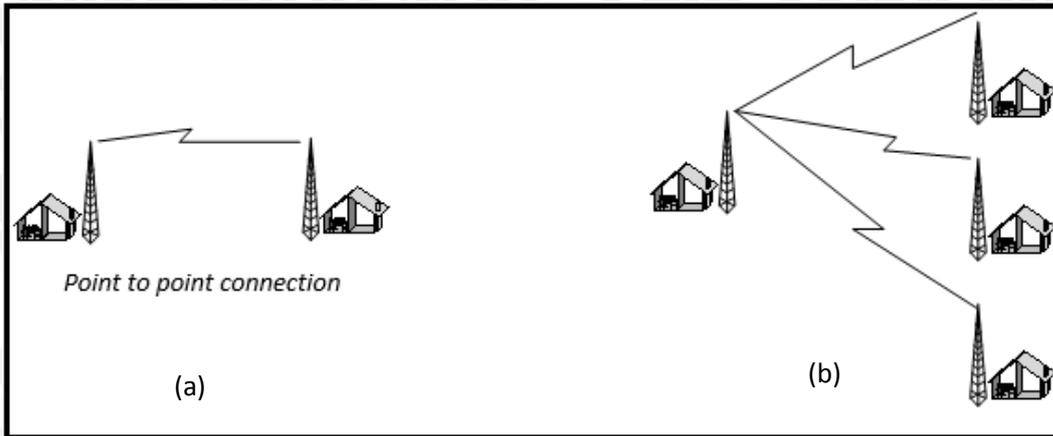
b. *Point to Multipoint*

Point to Multipoint adalah jaringan yang menghubungkan antara sisi pemancar dan sisi penerima terdapat satu perangkat pemancar dan banyak perangkat penerima. Pada Gambar 2.5, gambar a menjelaskan metode *point to point* dan gambar b merupakan penjelasan tentang metode *point to multipoint*. Karakteristik dasar standar WiMAX 802.16 dan 802.16-2004 dapat dilihat pada tabel 2.4.

Tabel 2.4 Karakteristik Dasar Standar WiMAX

Parameter	802.16	802.16-2004
Status	Desember 2001	Juni 2004
Frekuensi kerja	10 GHz-66 GHz	2 GHz-11 GHz
Aplikasi	<i>Fixed LOS</i>	<i>Fixed NLOS</i>
Arsitektur MAC	<i>Point-to-multipoint, mesh</i>	<i>Point-to-multipoint, mesh</i>
Skema transmisi	<i>Single carrier</i>	<i>Single carrier, 256 OFDM atau 2048 OFDM</i>
Modulasi	QPSK, 16QAM, 64QAM	QPSK, 16QAM, 64QAM
Laju data kotor	32 Mbps-134.4 Mbps	1 Mbps-75 Mbps
Multiplexing	<i>Burst TDM/TDMA</i>	<i>Burst TDM/TDMA/OFDM</i>
Duplexing	TDD dan FDD	TDD dan FDD
Channel Bandwidth	20 MHz, 25 MHz, 28 MHz	1.75 MHz, 3.5 MHz, 7 MHz, 14 MHz, 1.25 MHz, 5 MHz, 10 MHz, 15 MHz, 8.75 MHz

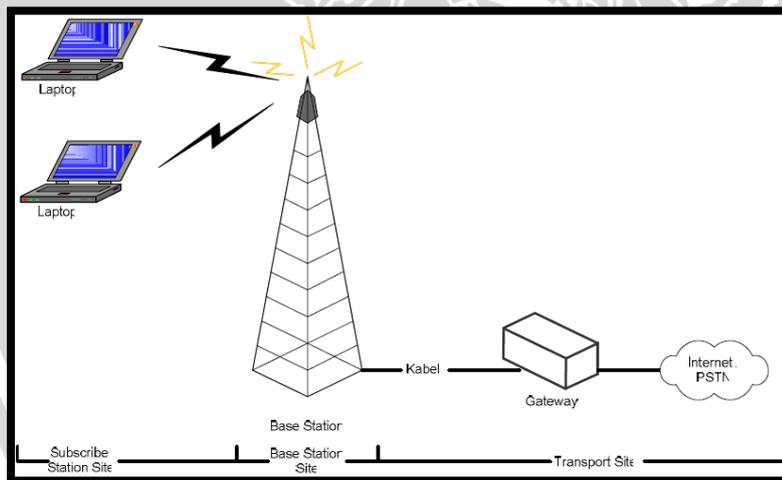
Sumber: Jeffrey G Andrews (2007: 36)



Gambar 2.5 Metode Topologi Jaringan WiMAX
(Sumber: www.wimaxforum.org/topologi)

2.4.2 Konfigurasi Jaringan WiMAX

Sistem WiMAX terdiri dari beberapa bagian, yaitu *Base Station* (BS) di sisi pusat dan *Subscriber Station* (SS) atau *Customer Premise Equipment* (CPE) di sisi pelanggan. Konfigurasi jaringan WiMAX dapat dilihat pada gambar 2.6.



Gambar 2.6 Konfigurasi Jaringan WiMAX.
(Sumber: Wibisono, 2009)

Dalam gambar 2.6, digambarkan bahwa konfigurasi WiMAX terdiri dari:

1. *subscriber station* (SS) site, bagian ini terdiri dari *Customer Premise Equipment* (CPE) atau dapat berupa laptop. Bagian ini berfungsi sebagai peralatan yang digunakan oleh *user/pengguna*.
2. *base station* (BS) site, bagian ini terdiri dari *base station* (BS). Bagian ini berfungsi untuk menghubungkan *subscriber station* (SS) site dengan *transport site*.

3. *transport site* (bagian *backend*), bagian ini terdiri dari jaringan *internet/PSTN*. Bagian ini berfungsi untuk menghubungkan *base station* dengan *internet*.

2.5 Perangkat Lunak

2.5.1 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 2.7 merupakan logo perangkat lunak *wireshark*. *Wireshark* memiliki fungsi sebagai *packet sniffer*, yang memiliki 3 proses, antara lain:

1. Capture

Wireshark menangkap paket data yang melewati adapter kabel jaringan maupun *wireless* pada pengguna.

2. Decode

Decode adalah proses mengubah suatu informasi pada *protocol* tertentu menjadi sebuah informasi bentuk lain yang diinginkan.

3. Analyze

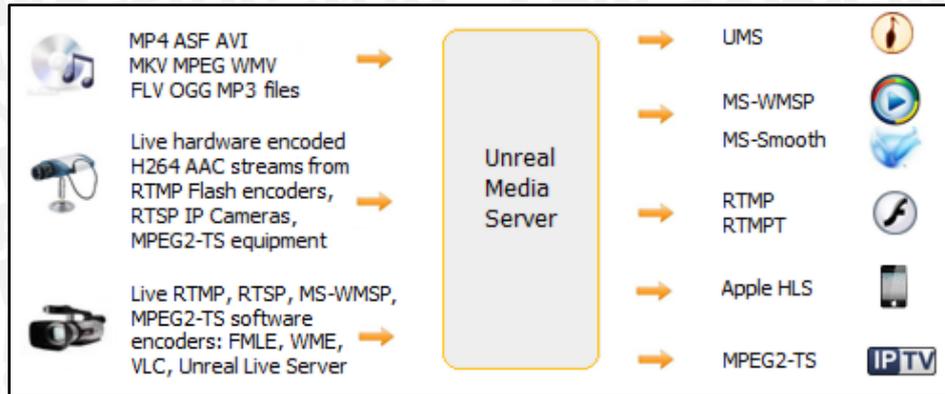
Analyze menampilkan beberapa parameter terkait performansi maupun karakteristik paket data.



Gambar 2.7 Logo *Wireshark*
(Sumber: www.wireshark.org)

2.5.2 Unreal Media Server dan Unreal Media Player

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



Gambar 2.8 Unreal Media Server
(Sumber: unmediaserver.net)

Berdasarkan gambar 2.8, *Unreal Media Server* mampu menyiarkan format multimedia. Format-format multimedia yang didukung antara lain: MP4, ASF, AVI, MPEG, WMV, FLV, OGG, MP3, 3GP, dan MOV (*umediaserver.net*).

2.5.3 Solarwind

Solarwind adalah perangkat lunak yang berfungsi untuk mengirimkan satu paket atau lebih dalam jaringan *server*. *Solarwind* digunakan untuk mengetahui performansi pengujian beban maksimal pada jaringan.



Gambar 2.9 Logo Solarwinds
(Sumber: solarwind.com)

2.6 Parameter Kinerja IPTV

Parameter kinerja merupakan ukuran atau acuan yang digunakan untuk mengetahui kinerja jaringan dan untuk mengetahui kinerja dari jaringan VLAN WIMAX, terdapat beberapa parameter yang akan digunakan untuk mengukur kualitas kinerja layanan, beberapa parameter performansi yang akan digunakan dalam skripsi ini antara lain *signal to noise ratio*, kapasitas kanal, *bit rate* pada modulasi, *energy bit to noise*, *bit error rate*. Subbab berikut adalah penjelasan untuk masing-masing parameter tersebut.

2.6.1 Perhitungan *Signal to Noise Ratio*

Signal to Noise Ratio merupakan perbandingan antara daya yang dipancarkan *transmitter* terhadap *noise* yang timbul pada saat proses *propagasi*. Besarnya SNR dinyatakan dalam persamaan 2-8 (Mischa Schwartz, 1994)

$$SNR = P_r - N_o \quad (2-8)$$

Keterangan :

SNR = *signal to noise ratio* (dB)

P_r = daya yang diterima (dB)

N_o = daya *noise* pada saluran transmisi (dB)

Dengan daya *noise* pada saluran transmisi dinyatakan dalam persamaan 2-9.

$$N_o = 10 \log kT + 10 \log B + NF \quad (2-9)$$

Keterangan :

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* (3,2 dB)

B = *bandwidth* (Hz)

Dari persamaan 2-8 diperoleh SNR_{system} dengan persamaan 2-10.

$$SNR_{\text{system}} = (1 - \alpha_{CP}) \times SNR \quad (2-10)$$

Keterangan:

α_{CP} = *Cyclic Prefiks*

2.6.2 Perhitungan Kapasitas Kanal

Kapasitas kanal merupakan salah satu parameter kinerja dari kanal *wireless*. Kapasitas kanal adalah banyaknya informasi yang bisa dilewatkan dalam sebuah kanal *frekuensi*, mempunyai satuan *bit per sekon* (bps). Semakin besar kapasitasnya kanal akan semakin baik. Perhitungan kapasitas kanal dinyatakan dengan persamaan 2-11 (*Shannon*).

$$C = B \log_2(1 + SNR) \quad (2-11)$$

C = kapasitas kanal sistem (bps)

B = *bandwidth* sistem (Hz)

SNR = *signal to noise ratio* sistem

2.6.3 Perhitungan *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, ditandai dengan banyaknya *bit* per simbol yang dikirim pada setiap jenis modulasi. Semakin besar *bit rate* maka semakin cepat pengiriman informasi. Perhitungan *bit rate* dinyatakan dengan persamaan 2-12 (Stallings, 2005).

$$R = f \times \log_2 n \quad (2-12)$$

Keterangan:

n = banyaknya *bit* pada suatu jenis modulasi

f = *bandwidth* kanal yang dipergunakan (Hz)

R = *bit rate* (bps)

2.6.4 Perhitungan *Energi Bit to Noise*

Energy bit to noise adalah perbandingan nilai *energy bit* data yang akan dikirimkan dengan nilai daya *noise* yang akan terjadi. Nilai *energy bit to noise* digunakan untuk menilai kualitas sinyal yang dikirim oleh pemancar. Nilai perbandingan *energy bit* terhadap *noise* adalah berikut (Stallings, 2005):

$$\left(\frac{E_b}{N_0}\right) = SNR \times \frac{B_{dB}}{R_{dB}} \quad (2-13)$$

Keterangan:

R_{dB} = $10 \log R$ (*bit rate* dalam dB)

B_{dB} = $10 \log B$ (*bandwidth* dalam dB)

2.6.5 Perhitungan *Bit Error Rate*

Bit Error Rate atau yang disebut juga *probabilitas bit* salah adalah peluang besarnya *bit* salah yang mungkin terjadi selama proses pengiriman data. Nilai dari *probabilitas bit* salah dapat diperoleh dari perhitungan yang dilakukan berdasarkan modulasi sinyal yang digunakan. Modulasi yang digunakan pada skripsi ini menggunakan modulasi 64 QAM.

Modulasi 64 QAM merupakan jenis modulasi QAM (*Quadrature Amplitude Modulation*) yang sedikit berbeda dari modulasi 16 QAM. Pada modulasi ini terdapat 6 *bit* pada tiap simbolnya, sehingga disebut juga dengan *hexabit*. Teknik 64-QAM merupakan teknik yang paling cepat mentransmisikan data pada teknologi WiMAX, 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):

$$P_{b,64QAM} = \frac{3}{2k} \operatorname{erfc} \left(\sqrt{\frac{k \cdot E_b}{10 N_0}} \right) \quad (2-14)$$

Keterangan:

$$\frac{E_b}{N_0} = \text{energy bit to noise (dB)}$$

$P_{b,64QAM}$ = Probabilitas bit salah/bit error rate pada modulasi 64 QAM

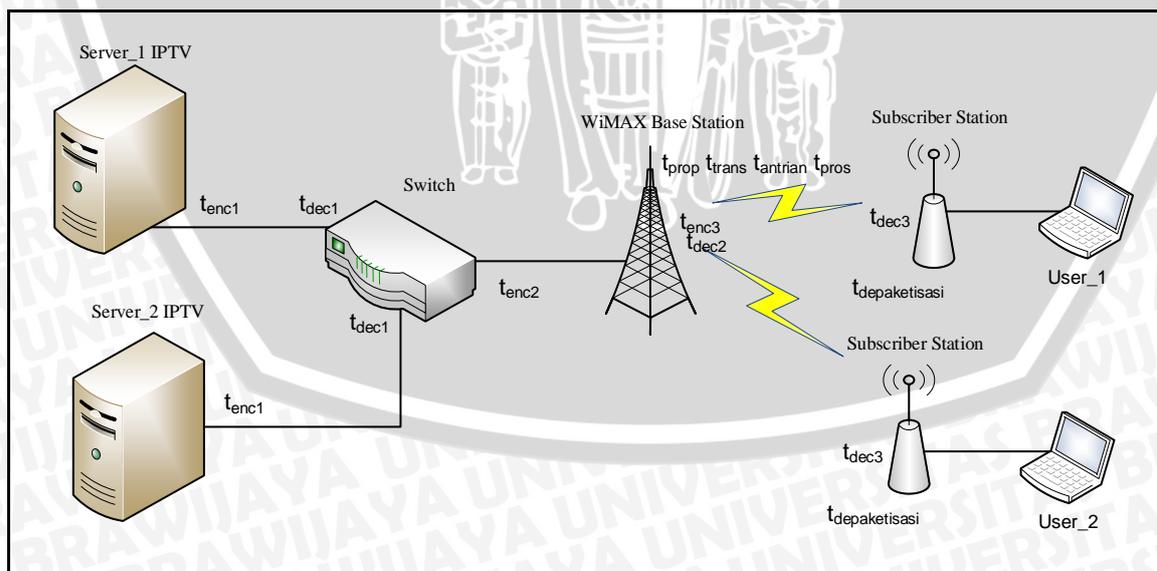
2.6.6 Delay End to End

Delay end to end merupakan penjumlahan beberapa macam komponen *delay* yang berbeda dan terjadi dalam perjalanan paket dari sumber sampai ke tujuan. Gambar 2.10 menunjukkan ilustrasi proses terjadi *delay end to end*. Tabel 2.5 merupakan pengelompokan *delay* berdasarkan ITU-T G.114.

Tabel 2.5 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



Gambar 2.10 *Delay End to End*
(Sumber: Mischa Schwartz, 1994)

Skripsi ini membatasi perhitungan pada kanal *downlink*, yakni dari *server* menuju *user* karena paket data aplikasi IPTV disiarkan dari *server* menuju *user* tanpa *user* mengirim data informasi menuju *server*. Perhitungan *delay end to end* dinyatakan dengan persamaan 2-15 (Mischa Schwartz, 1994).

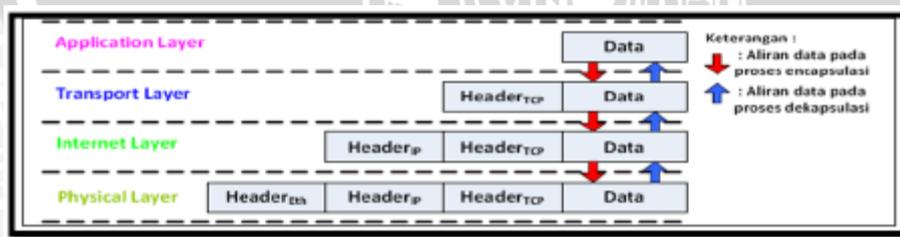
$$T_{end\ to\ end} = t_{proses} + t_t + t_p + t_w + t_{dec} \quad (2-15)$$

Keterangan:

- $t_{end\ to\ end}$ = Delay end-to-end (s)
 t_{proses} = Delay proses (s)
 t_t = Delay transmisi (s)
 t_p = Delay propagasi (s)
 t_w = Delay antrian (s)
 t_{dec} = Delay paketisasi dan 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 WiMAX terdiri dari *delay enkapsulasi* dan *delay dekapsulasi*. *Delay enkapsulasi* adalah waktu yang dibutuhkan untuk menambahkan *header* pada paket data, sehingga paket data tersebut dapat tepat sampai ke *host* tujuan. Sedangkan *delay dekapsulasi* adalah waktu yang dibutuhkan untuk melepaskan keseluruhan *header* dari sebuah paket data. Gambar 2.11 menunjukkan terjadi *delay proses enkapsulasi dan dekapsulasi*.



Gambar 2.11 Proses *Enkapsulasi* dan *Dekapsulasi* Paket Data
 (Sumber: <http://www.citap.com/document/tcp-ip/tcpip011.htm>)

Apabila *host* sumber ingin mengirim paket data ke *host* tujuan, maka proses yang terjadi adalah paket data tersebut dikirimkan ke *transport layer* dan mengalami perubahan format data yaitu penambahan *header* RTP dan UDP. Besarnya data yang ditambahkan *header* RTP dihitung menggunakan persamaan 2-16.

$$W_{message} = W_{data} + Header_{RTP} \quad (2-16)$$

Keterangan :

$W_{message}$ = panjang *message* (byte/paket)

$Header_{RTP}$ = panjang *header* RTP (12 byte)

Format paket data pada RTP ditunjukkan pada gambar 2.12



Gambar 2.12 Format Paket Data RTP.
(Sumber: Nuaymi, 2007)

Pada *layer transport* terjadi penambahan *header* UDP, sehingga panjang *segmen* data tidak boleh melebihi *Maximal Segmen Size* (MSS) yaitu sebesar 1460 *byte*, maka dilakukan proses *fragmentasi* yaitu pemecahan paket data agar sesuai dengan batasan jaringan, jumlah *segmen* hasil *fragmentasi* dihitung menggunakan persamaan 2-17.

$$N_{segmen} = \frac{W_{message}}{MSS} \quad (2-17)$$

Keterangan:

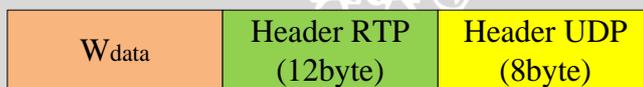
N_{segmen} = total *segmen* UDP yang dikirimkan (byte)

MSS = Maximum *Segmen* Size (1460 byte)

Setelah *segmen* difragmentasi, proses berikutnya adalah proses *enkapsulasi* yang dapat dihitung sesuai dengan persamaan 2-18

$$W_{segmen} = W_{message} + (N_{segmen} \times Header_{UDP}) \quad (2-18)$$

Format paket data pada TCP/UDP ditunjukkan pada gambar 2.14



Gambar 2.13 Format Paket Data TCP/UDP.
(Sumber: Nuaymi, 2007)

Dari *transport layer* (*layer* 4), *segmen* kemudian diteruskan ke *internet layer* (*layer* 3). Di *internet layer* dengan menggunakan *protocol* IP, *segmen* UDP dienkapsulasi menjadi *datagram* IP. Panjang *datagram* IP ditentukan dengan persamaan 2-19.

$$W_{datagram} = W_{segmen} + Header_{IP} \quad (2-19)$$

Keterangan :

$W_{datagram}$ = panjang *datagram* IP (byte/paket)

W_{segmen} = panjang *segmen* UDP (byte/paket)

$Header_{IP}$ = panjang *header* IP (20 byte/paket)

Format paket data pada IP ditunjukkan pada gambar 2.15

W_{data}	Header RTP (12 byte)	Header UDP (8 byte)	Header IP (20 byte)
------------	-------------------------	------------------------	------------------------

Gambar 2.14 Format Paket Data IP.
(Sumber: Nuaymi, 2007)

Kemudian *datagram* IP dienkapsulasi dengan *header* pada *network header layer* (layer 2) dengan menggunakan *protocol ethernet*, *datagram* IP dienkapsulasi menjadi *frame ethernet*. Panjang *frame ethernet* ditentukan dengan persamaan 2-20.

$$W_{frame\ ethernet} = W_{datagram} + Header_{ethernet} \quad (2-20)$$

Keterangan :

W_{frame} = panjang *frame Ethernet* (byte)

$W_{datagram}$ = panjang *datagram* IP (byte)

$Header_{IP}$ = panjang *header Ethernet* (14byte)

Format paket data pada *Ethernet* ditunjukkan pada gambar 2.15.

W_{data}	Header RTP (12 byte)	Header UDP (8 byte)	Header IP (20 byte)	Header Ethernet (14 byte)	CRC (4 byte)
------------	-------------------------	------------------------	------------------------	------------------------------	-----------------

Gambar 2.15 Format Paket Data *Ethernet*.
(Sumber: Nuaymi, 2007)

Dari *Ethernet*, *frame Ethernet* dikirimkan ke *layer MAC* WiMAX dan mengalami penambahan *header MAC*. Maka W_{frame} total dapat dicari dengan persamaan 2-21.

$$W_{frame\ total} = W_{frame\ ethernet} + Header_{MAC} + CRC + Header_{VLAN} \quad (2-21)$$

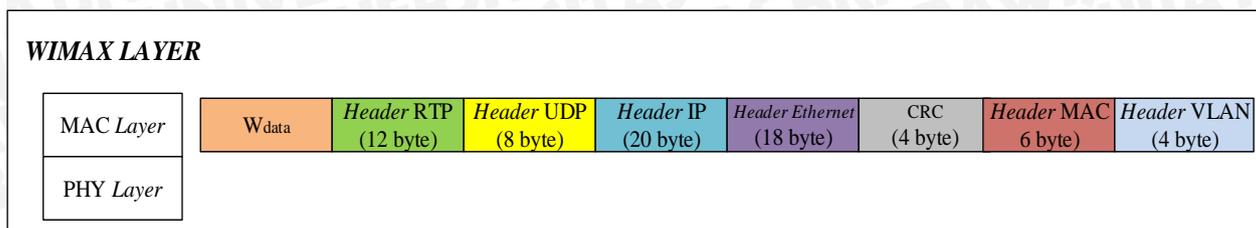
Keterangan:

$Header_{MAC}$ = panjang *header* pada MAC (6 byte)

CRC = panjang CRC (4 byte)

$Header_{VLAN}$ = panjang *header* (4 byte)

Format paket data pada *layer* MAC WiMAX ditunjukkan pada gambar 2.16



Gambar 2.16 Format Paket Data MAC WiMAX.
(Sumber: Nuaymi, 2007)

Nilai dari $W_{frame\ total}$ digunakan untuk mendapatkan *delay enkapsulasi* dan *delay dekapsulasi*. Persamaan *delay enkapsulasi* dinyatakan dengan persamaan 2-22.

$$t_{enc} = \frac{W_{frame\ total}}{c} \times 8 \quad (2-22)$$

Keterangan :

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

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

C = kecepatan pemrosesan data (bps)

Sedangkan *delay dekapsulasi* dinyatakan dengan persamaan 2-23.

$$t_{dec} = \frac{W_{frame\ total}}{c} \times 8 \quad (2-23)$$

Dengan :

t_{dec} = *delay enkapsulasi* (s)

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

C = kecepatan pemrosesan data (bps)

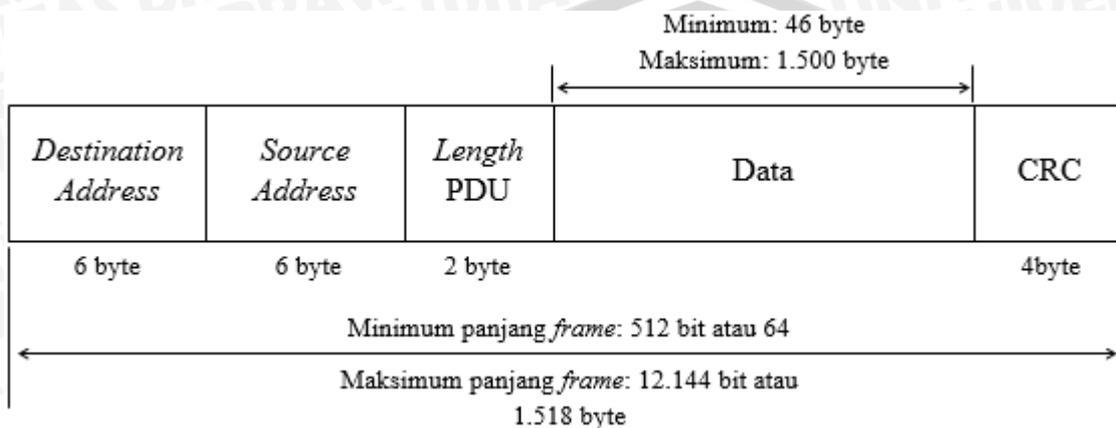
Dari hasil perhitungan *delay enkapsulasi* dan *delay dekapsulasi* tersebut, maka akan didapatkan nilai *delay proses* dengan persamaan 2-24.

$$t_{proses} = t_{enc} + t_{dec} \quad (2-24)$$

Jadi, *delay proses* merupakan jumlah dari *delay enkapsulasi* dan *delay dekapsulasi*

b) Delay Transmisi

Delay transmisi adalah waktu yang dibutuhkan untuk meletakkan sebuah paket data ke media transmisi. *Delay transmisi* ini dipengaruhi oleh ukuran paket dan kapasitas kanal media transmisi. Gambar 2.17 merupakan nilai panjang *frame* yang digunakan pada sistem jaringan. *Delay transmisi* dinyatakan dengan persamaan 2-25 (Forouzan,2007)



Gambar 2.17 Minimum dan Maksimum Panjang *Frame*
(Sumber: Forouzan, 2007)

$$t_{trans} = \frac{W_{frame\ total}}{C} \times 8 \quad (2-25)$$

Keterangan:

t_{trans} = *Delay transmisi* (s)

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

C = kapasitas kanal (bps)

c) Delay Propagasi

Delay propagasi adalah waktu yang dibutuhkan untuk merambatkan gelombang radio pada media transmisi. Kecepatan *propagasi* bergantung kepada karakteristik fisik koneksi dan jarak antara pengirim dan penerima. *Delay propagasi* dinyatakan dengan persamaan 2-26.

$$t_{prop} = \frac{d}{c} \quad (2-26)$$

Keterangan:

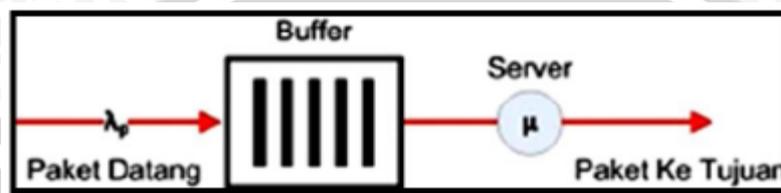
t_{prop} = *delay propagasi* (s)

d = jarak antara pengirim dan penerima (m)

c = cepat rambat media *propagasi* (3×10^8 m/s)

d) Delay Antrian

Delay antrian adalah waktu yang dibutuhkan data selama berada dalam antrian untuk ditransmisikan. *Delay* antrian dapat dihitung dengan menggunakan model antrian M/M.1. M pertama menunjukkan kedatangan *Poisson*, M kedua berarti distribusi waktu pelayanan eksponensial, dan 1 menunjukkan bahwa jumlah pelayannya adalah tunggal. Besarnya *delay* antrian yang terjadi pada *router* ditentukan dengan Persamaan 2-27 (Mischa Schwartz, 1987: 42).



Gambar 2.18 Model Antrian M/M/S
(Sumber: Mischa Schwartz, 1987: 42)

$$t_W = \frac{1}{\mu} + \frac{\lambda_W}{\mu^2(1-\rho)} \quad (2-27)$$

Keterangan:

t_W = *delay* antrian (s)

μ = kecepatan pelayanan paket (paket/s)

λ_W = kecepatan kedatangan paket (paket/s)

ρ = faktor *utiliti*

Kecepatan pelayanan paket ditentukan dengan persamaan 2-28.

$$\mu = \frac{C}{L} \quad (2-28)$$

Keterangan:

μ = kecepatan pelayanan paket (paket/s)

C = kapasitas kanal (bps)

L = panjang paket (*bit*)

Sedangkan untuk menentukan kecepatan kedatangan paket digunakan persamaan 2-29.

$$\lambda_W = \mu \times \rho \quad (2-29)$$

Keterangan:

λ_W = kecepatan kedatangan paket (paket/s)

μ = kecepatan pelayanan paket (paket/s)

ρ = faktor *utiliti*

e) Delay Paketisasi dan Depaketisasi

Delay paketisasi adalah waktu yang dibutuhkan untuk mempacketkan 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 paketisasi* dinyatakan dengan persamaan 2-30.

$$t_{\text{paketisasi}} = \frac{L_{\text{paket}}}{C} \quad (2-30)$$

Keterangan:

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

C = kapasitas kanal transmisi (bps)

Sedangkan untuk *delay depaketisasi* dinyatakan dengan persamaan 2-31.

$$t_{\text{depaketisasi}} = \frac{L_{\text{paket}}}{C} \quad (2-31)$$

Keterangan:

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

C = kapasitas kanal transmisi (bps)

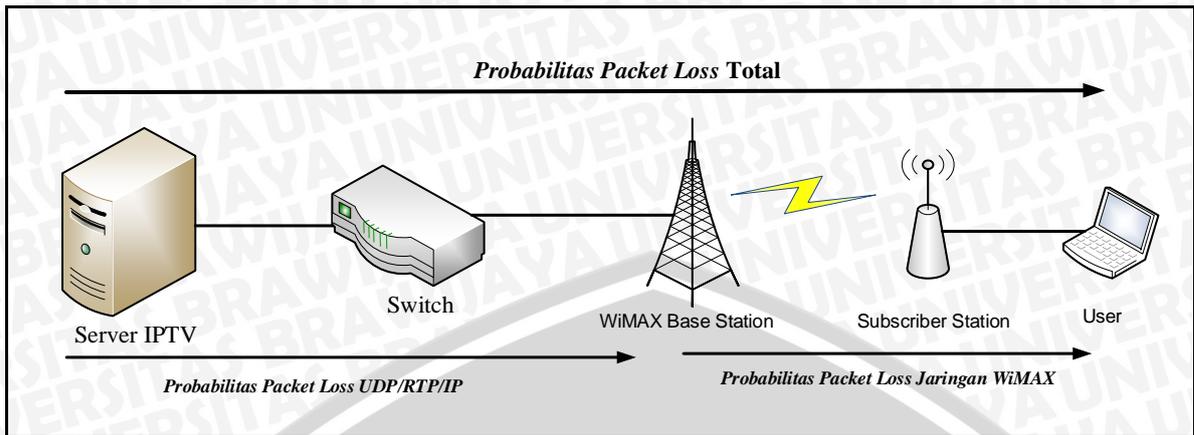
2.6.7 Packet Loss

Packet Loss adalah jumlah paket IP yang hilang selama proses transmisi dari *source* menuju *destination*. Beberapa penyebab terjadinya *packet loss* yaitu *congestion*, *node* yang bekerja melebihi kapasitas *buffer* dan *memory* yang terbatas pada *node*. Gambar 2.19 menunjukkan ilustrasi terjadinya *probabilitas packet loss* pada jaringan WiMAX. Berdasarkan ITU-T G.1010, aplikasi IPTV memiliki persyaratan *packet loss* yang berbeda-beda berdasarkan media yang dikirimkan. Berikut adalah tabel 2.6 yang menunjukkan syarat *packet loss* berdasarkan ITU-T G.1010.

Tabel 2.6 Standar *Packet Loss* ITU-T G.1010

Medium	Application	Degree of Symetry	Information Loss
Audio	Conversational Voice	Two-way	<3% Packet Loss Ratio
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



Gambar 2.19 *Probabilitas Packet Loss*
(Sumber: Pritchard, Et.al, 1993)

Probabilitas packet loss total pada WiMAX ditentukan berdasarkan *packet loss* pada jaringan yang berbasis *protocol* UDP/RTP/IP seperti ditunjukkan pada persamaan 2-32 (Pritchard, Et.al, 1993).

$$\rho_{tot} = 1 - [(1 - \rho_{net})(1 - \rho_{TCP/IP})] \quad (2-32)$$

Keterangan:

ρ_{tot} = *Probabilitas packet loss total*

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

ρ_{IPTV} = *Probabilitas packet loss model layer UDP/RTP/IP*

Besarnya *probabilitas packet loss* dalam persen dirumuskan dalam persamaan 2-33.

$$\rho_{tot}(\%) = \rho_{tot} \times 100 \% \quad (2-33)$$

2.6.8 Throughput

Throughput adalah rata-rata jumlah data yang diterima dalam keadaan benar setelah melalui kanal komunikasi seperti kanal fisik, kanal logik atau telah melalui beberapa *network node*. *Throughput* juga dapat diartikan sebagai kecepatan maksimal jaringan saat tidak ada data yang hilang pada saat proses transmisi. Nilai *throughput* dihitung pada proses transmisi sinyal pada jaringan WiMAX hingga informasi diterima pada *Base Station* (BS). Perhitungan *throughput* dinyatakan dengan persamaan 2-34 (Stallings, 2005).

$$\lambda = \frac{(1-\rho)}{t_{trans}[1+(\alpha-1)\rho]} \quad (2-34)$$

Keterangan :

λ = *throughput* (paket/s)

ρ = *probabilitas packet loss*

t_{trans} = *delay transmisi* (s)

Simbol α merupakan konstanta *propagasi* dengan persamaan 2-35.

$$\alpha = \left(3 + \frac{2t_{prop}}{t_{paketisasi}} \right) \quad (2-35)$$

Keterangan :

α = konstanta *propagasi*

t_{prop} = *delay propagasi* (s)

$t_{paketisasi}$ = *delay paketisasi* (s)

