

BAB II

TINJAUAN PUSTAKA

2.1 *Worldwide Interoperability for Microwave Access (WiMAX)*

Worldwide Interoperability for Microwave Access (WiMAX) merupakan teknologi akses nirkabel pita lebar (*Broadband Wireless Access*) yang dapat digunakan dalam kondisi LOS (*Line of Sight*) maupun NLOS (*Non Line of Sight*). Selain itu, WiMAX dihadirkan untuk menyediakan teknologi transfer data kecepatan tinggi. Kecepatan transfer datanya dapat mencapai hingga 75 Mbps serta jarak jangkauan hingga 50 km. Melalui *Institute of Electrical and Electronic Engineers (IEEE)*, WiMAX diputuskan menjadi standar kerja 802.16 agar dapat dikembangkan menjadi teknologi yang mampu menyediakan layanan jaringan publik. Setelah melalui beberapa perkembangan, WiMAX memiliki beberapa *sub* standar yang dapat dilihat pada Tabel 2.1.

Tabel 2.1

Standar WiMAX IEEE 802.16d dan 802.16e

<i>Specification</i>	802.16	802.16d HiperMAN	802.16e
<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>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>	QPSK, 16 QAM and 64 QAM	OFDM 256 FFT, QPSK, 16 QAM, 64 QAM	Scalable OFDMA, 128- 2.048 FFT/BPSK, QPSK, 16 QAM, 64 QAM
<i>Mobility</i>	<i>Fixed</i>	<i>Fixed</i>	<i>Nomadic/mobile</i>
<i>Channel Bandwidth</i>	20 MHz, 25 MHz and 28 MHz	(1.75 – 20) MHz	(1.75 – 20) MHz

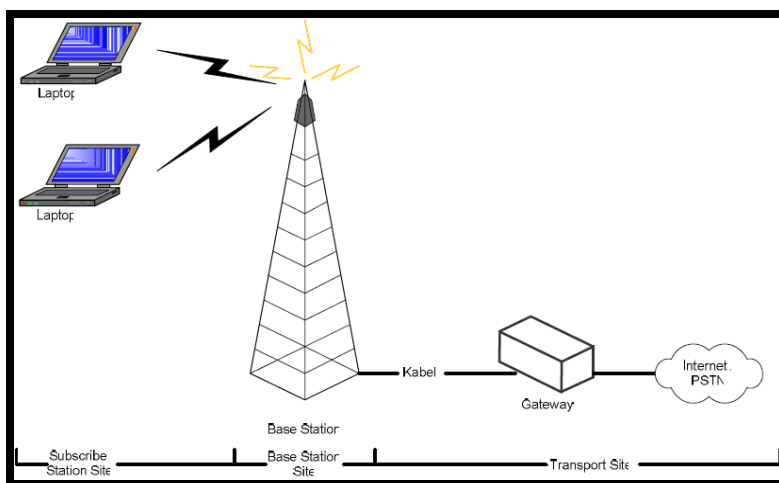
Sumber: Zerihun Abate, WiMAX RF System Engineering (2009)

2.1.1 Konfigurasi Jaringan WiMAX

Konfigurasi jaringan WiMAX secara umum tidak berbeda jauh dengan jaringan WLAN. Sistem jaringan WiMAX tersusun dari tiga komponen utama yaitu

1. *Base Station (BS)*, merupakan perangkat *transceiver (transmitter dan receiver)*
2. Antena, suatu pemancar gelombang elektromagnetik yang dihubungkan ke BS.
3. *Subscriber Station (SS)* adalah perangkat receiver yang dapat disambungkan ke beberapa *Customer Premise Equipment (CPE)*

Komponen tersebut nantinya akan ditempatkan di bagian yang telah ditentukan. Penempatan komponen tersebut bisa dilihat pada Gambar 2.1.berikut.



Gambar 2.1 Konfigurasi Jaringan WiMAX.

Sumber: Wibisono (2009)

Pada Gambar 2.1 ditunjukkan bahwa bagian tersebut terdiri dari :

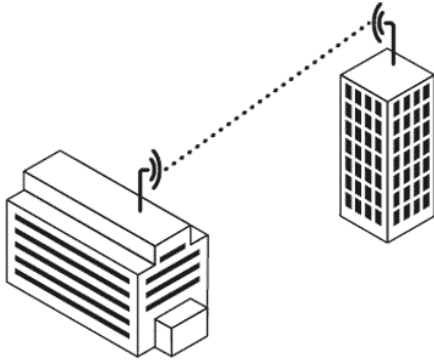
1. *Subscriber Station (SS) site*, bagian ini terdiri dari *Customer Premise Equipment (CPE)* atau dapat berupa laptop. Bagian ini berfungsi sebagai peralatan yang digunakan oleh pengguna.
2. *Base Station (BS) site*, bagian ini terdiri dari *base station (BS)*. Bagian ini berfungsi untuk menghubungkan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

2.1.2 Topologi Jaringan WiMAX

Seiring perkembangannya, WiMAX memiliki beberapa topologi jaringan yaitu *point to point*, *point to multipoint* dan *mesh*.

a. *Point To Point*

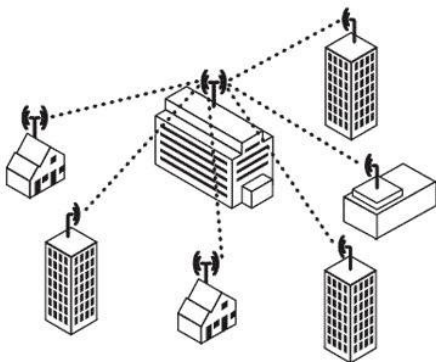
Pada topologi *Point-To-Point*, jaringan hanya memiliki sebuah *Base Station* dan sebuah *Subscriber Station*, sehingga jaringan hanya memiliki sebuah koneksi tunggal. Topologi jaringan *Point-To-Point* dapat dilihat pada Gambar 2.2



Gambar 2.2 Topologi Jaringan *Point-To-Point*.
Sumber: wi-man.net

b. *Point-To-Multipoint*

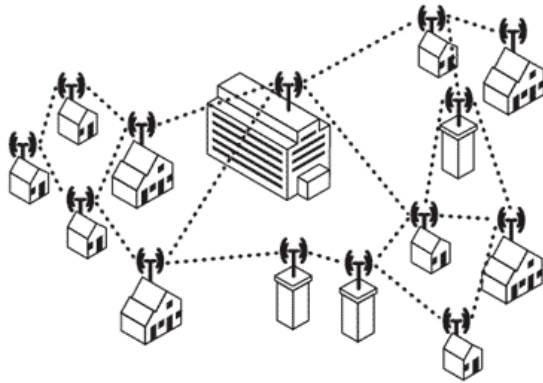
Pada topologi *Point-To-Multipoint*, jaringan memiliki sebuah *Base Station* dan beberapa *Subscriber Station* yang terhubung pada *Base Station*. Topologi jaringan *Point-To-Multipoint* dapat dilihat pada Gambar 2.3



Gambar 2.3 Topologi Jaringan *Point-To-Multipoint*.
Sumber: wi-man.net

c. *Mesh*

Topologi *mesh* merupakan topologi gabungan antara topologi *point to point* dan topologi *point to multipoint*. Topologi jaringan *Point-To-Multipoint* dapat dilihat pada Gambar 2.4



Gambar 2.4 Topologi Jaringan *mesh*.
Sumber: wi-man.net

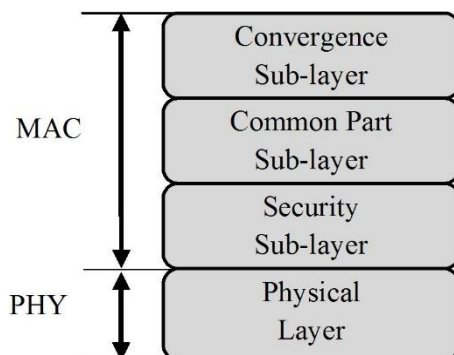
2.1.3 Prinsip Kerja WiMAX

Secara umum prinsip kerja WiMAX adalah sebagai berikut:

1. Pengguna yang terhubung jaringan mengirimkan data dari *Subscriber Station (SS)* ke *Base Station (BS)* melalui media gelombang radio.
2. *Base Station* akan menerima sinyal dan mengirimkan pesan melalui *wireless* atau kabel ke *Switching Center* melalui protokol IEEE 802.1.
3. *Switching center* akan mengirimkan pesan ke *Internet Service Provider* atau *Public Switched Telephone Network (PSTN)* melalui kabel. (Gunawan Wibisono, 2009).

2.1.4 Struktur Layer WiMAX

Menurut standar IEEE.802.16, WiMAX dikhususkan untuk pengembangan teknologi pada layer fisik (PHY) dan layer data link (MAC). *Physical layer* berfungsi sebagai transportasi data bersifat fundamental yang bertujuan untuk mengalirkan data di lapisan fisik. Sedangkan Medium Access Control (MAC) layer berfungsi sebagai penterjemah protokol-protokol yang ada di atasnya seperti ATM, Ethernet dan IP. Susunan layer WiMAX dapat dilihat pada Gambar 2.5



Gambar 2.5 Referensi model Layer WiMAX.
Sumber: Mohammad Mehdi Gilanian Sadeghi (2013)

2.1.5 *Orthogonal Frequency-Division Multiplexing (OFDM)*

OFDM (*Orthogonal Frequency Division Multiplexing*) merupakan sebuah teknik transmisi dengan beberapa frekuensi (*multicarrier*) yang saling tegak lurus (*orthogonal*). Pada prinsipnya, teknik OFDM hampir sama dengan FDM (*Frequency Division Multiplexing*) yaitu membagi lebar pita (*Bandwidth*) yang ada kedalam beberapa kanal. Namun teknik OFDM membagi kanal tersebut dengan lebih efisien dibanding sistem FDM. Karena masing-masing frekuensi sudah saling tegak lurus (*orthogonal*) sehingga terjadi *overlap* antar frekuensi yang bersebelahan, maka tidak diperlukan *guard band*.

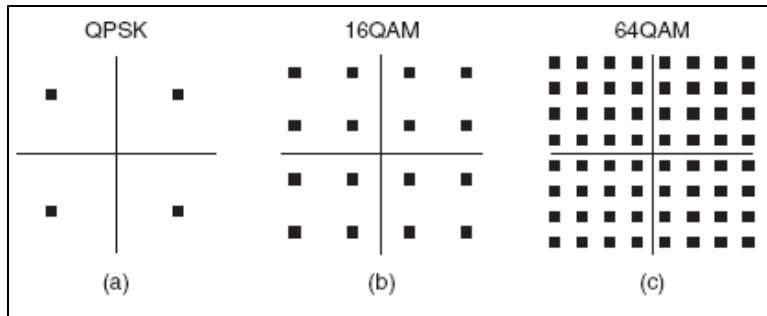
Prinsip kerja OFDM adalah deretan data informasi yang akan dikirim dikonversikan ke dalam bentuk paralel, sehingga bila *bit rate* semula adalah R, maka bit rate di tiap-tiap jalur paralel adalah R/M dimana M adalah jumlah jalur paralel (sama dengan jumlah *sub-carrier*). Setelah itu, modulasi dilakukan pada tiap-tiap *sub-carrier*. Kemudian sinyal yang telah termulasi tersebut diaplikasikan ke dalam *Inverse Discrete Fourier Transform* (IDFT), untuk pembuatan simbol OFDM. Penggunaan IDFT ini memungkinkan pengalokasian frekuensi yang saling tegak lurus. Setelah itu simbol-simbol OFDM dikonversikan lagi ke dalam bentuk serial, dan kemudian sinyal dikirim. Pada OFDM, frekuensi-frekuensi *multicarrier* tersebut saling tegak lurus, yang berarti bahwa *crosstalk* di antara *sub-channels* dihilangkan dan *inter-carrier guard bands* tidak diperlukan.

Pada OFDM, sinyal didesain sedemikian rupa agar *orthogonal*, sehingga jika tidak ada distorsi pada jalur komunikasi yang menyebabkan ISI (*intersymbol interference*) dan ICI (*intercarrier interference*), maka setiap *subchannel* akan bisa dipisahkan stasiun penerima dengan menggunakan DFT. Solusi yang termudah adalah dengan menambah jumlah *subchannel* sehingga periode simbol menjadi lebih panjang, dan distorsi bisa diabaikan jika dibandingkan dengan periode simbol. Keunggulan OFDM adalah tingginya tingkat efisiensi dalam pemakaian frekuensi.

2.1.6 *Quadrature Amplitude Modulation (QAM)*

Teknik modulasi QAM terdiri dari sinyal amplitudo dan sinyal fasa yang berubah mengikuti perubahan amplitudo sinyal informasi, maka dipresentasikan sebagai besaran amplitudo dan pergeseran fasa. QAM memiliki jumlah minimum bit per simbolnya sebanyak 2 bit, namun QAM memiliki berbagai macam teknik lainnya seperti 4-QAM, 8-QAM, 16-QAM, 32-QAM, 64-QAM, 128-QAM, dan 256-QAM, dimana penggunaanya diatur berdasarkan banyaknya jumlah bit dalam satu simbol. Standar yang digunakan oleh

WiMAX 802.16 adalah 16-QAM dan 64-QAM. Konstelasi pada Gambar 2.8 menjelaskan hubungan antara amplitudo dan fasa dengan bit yang telah ditentukan.



Gambar 2.6 Konstelasi QAM.
Sumber: Erik Dahlman (2008)

2.1.7 Scheduling

Teknologi WiMAX dapat menjalankan *Scheduling* dengan berbagai kebutuhan aplikasi. Sebagai contoh aplikasi *streaming* dan *conferencing* memerlukan latency yang rendah tetapi masih bisa mentolerir beberapa *error rate*. Sebaliknya aplikasi-aplikasi data pada umumnya sangat sensitif terhadap *error rate*, sedangkan faktor latency bukan menjadi pertimbangan kritis. Perubahan parameter *Scheduling* bisa diminta oleh SS ke BS dengan sambungan masih tetap terjaga. Berdasarkan jenisnya, *Scheduling* pada WiMAX ini dapat dikelompokkan menjadi empat jenis yaitu *Unsolicited Grant Service* (UGS), *Real Time Packet Service* (rtPS), *Non-Real Time Packet Service* (nrtPS) dan *Best Effort* (BE).

a. Real Time Polling Service (rtPS)

Efektif untuk layanan yang sensitif terhadap *throughput* dan *delay*, garansi *bitrate* dan syarat *delay* telah ditentukan. Jenis layanan rtPS dirancang untuk mendukung aliran *real-time* data terdiri dari paket data berukuran *variabel* yang dikeluarkan pada interval waktu periodik.

b. Non-Real Time Polling Service (nrtPS)

Untuk pelanggan yang membutuhkan *bandwidth* yang besar dan efektif untuk aplikasi yang membutuhkan *throughput* yang baik. Layanan yang cocok untuk tipe *Scheduling* ini adalah *File Transfer* dan *E-mail*.

c. Best Effort (BE)

Best Effort adalah *Scheduling* yang digunakan jika masalah kecepatan data dan *delay* tidak terlalu diperhatikan namun sensitif terhadap *error rate*. Tipe *Scheduling* ini cocok untuk layanan *Web Browsing* dan *File Transfer*.

d. *Unsolicited grant service* (UGS)

Jenis *Scheduling* UGS dirancang untuk mendukung aliran data *real-time* yang terdiri dari paket data dengan ukuran tetap yang dikeluarkan pada interval periodik dengan *delay* minimum. Biasanya digunakan untuk menyediakan layanan yang memerlukan bit rate konstan seperti VoIP.

2.2 Video Streaming

Video streaming merupakan jenis media *streaming* yang mampu mengirim data sebuah *file* video menuju *user* secara terus menerus *via* jaringan internet. Data yang dikirim dapat disaksikan langsung tanpa menunggu *download* dari data tersebut selesai.

Video Streaming bekerja sesuai dengan prinsip *data streaming*, dimana semua data *file* data video dikompres dan dikirim dalam beberapa paket data dengan menggunakan protokol RTSP (*Real Time Streaming Protocol*). Data *file* yang akan dikirim nantinya diteruskan menuju *streaming server* setelah itu menuju *user*. Agar *user* dapat mengakses suatu *file* video, *Video Streaming* membutuhkan sebuah *Video Player* yang memiliki kemampuan memproses menggunakan protokol RTSP.

Menurut alur transmisi dari server menuju *user*, terdapat dua jenis tipe *video streaming*, yaitu:

1. *Live Streaming*, dimana video ditayangkan secara langsung (*Live*).
2. VoD (*Video On Demand*) dimana video yang akan ditampilkan sudah terlebih dahulu direkam (*pre-recorded*) atau disimpan dalam *server*.

2.2.1 Video On Demand (VoD)

Video on Demand merupakan salah satu jenis layanan dari *video streaming* dimana *user* dapat memilih video yang akan mereka tayangkan pada *device* mereka. Dengan menggunakan layanan *Video on Demand*, *user* dapat memilih kapan saat untuk memulai dan mengakhiri *streaming video*. Agar *file* video dapat disaksikan oleh *user* secara cepat dan tanpa interupsi, maka dibutuhkan koneksi internet yang stabil dengan bandwidth yang besar.

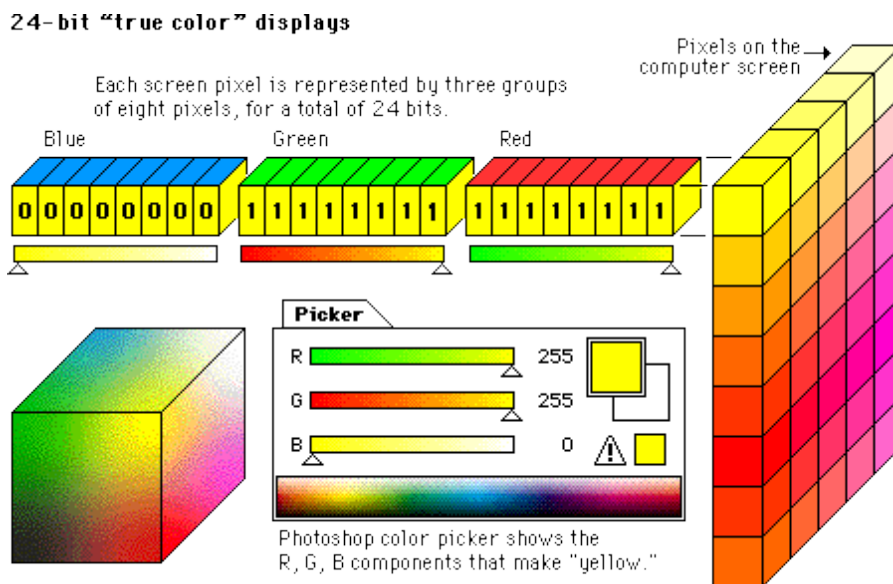
Terdapat beberapa parameter komponen *Video on Demand* terdiri dari sebagai berikut *Frame per Second*, *Video Bitrate*, *Resolution*,

1. *Frames per Second* (Fps) - merupakan satuan yang mengukur tampilan yang terjadi per detik.
2. *Video Bitrate* - merupakan nilai pengukuran dari bit yang dikirimkan per waktu tertentu.

3. *Resolution* - menunjukkan ukuran Gambar yang ditampilkan pada layar resolusi terdiri dari beberapa *pixel*. *Pixel* memiliki ukuran kedalaman. Semakin besar nilai kedalaman *pixel* menyebabkan jumlah bit yang digunakan untuk merepresentasikan 1 *pixel* semakin banyak dan ketajaman serta kejernihan Gambar semakin baik. Ukuran kedalaman *pixel* Gambar, antara lain: 8 bit (256 warna), 16 bit (65.536 warna) dan 32 bit (16.777.216 warna). Dalam skripsi ini kedalaman pixel yang digunakan adalah 32 bit.

2.2.2 Proses Data Menjadi Video

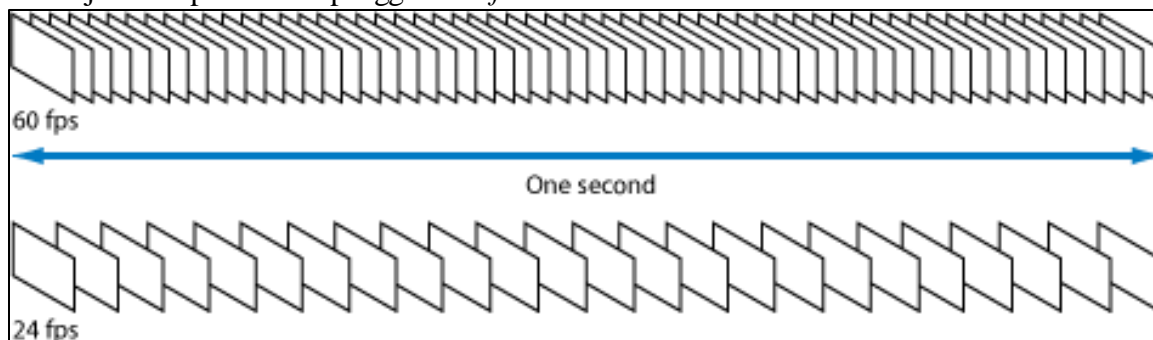
Video merupakan sebuah layanan multimedia dimana hasil tayangan merupakan kumpulan gambar (*frame*) yang diurutkan. Tampilan video juga dipengaruhi oleh resolusi gambar yang akan ditayangkan dimana resolusi video diwakilkan oleh lebar *frame* tersebut dalam satuan pixel. Selain itu, Tampilan *frame* dipengaruhi oleh warna dari sebuah pixel yang disebut *Color Depth* dengan satuan BPP (Bit Per Pixel). *Color Depth* memiliki beberapa macam tingkatan warna yaitu 8-bit, 16-bit, 24-bit, 32-bit dan 64-bit, semakin besar ukuran bit suatu pixel maka semakin detail dan bervariasi pula warna yang dihasilkan. Warna tersebut akan menentukan seberapa besar ukuran data dari sebuah *frame* yang akan ditampilkan dalam suatu video. Gambar 2.7 akan menunjukkan contoh bagaimana suatu data warna diubah menjadi sebuah *pixel*.



Gambar 2.7 Proses data warna ditampilkan menjadi sebuah pixel

Kumpulan pixel yang ada kemudian dijadikan suatu *frame*. Lalu *frame* tersebut akan disusun dan diurutkan menjadi sebuah video. Susunan *frame* tersebut diurutkan sesuai dengan *frame rate* yang digunakan, semakin besar *frame rate* yang digunakan maka

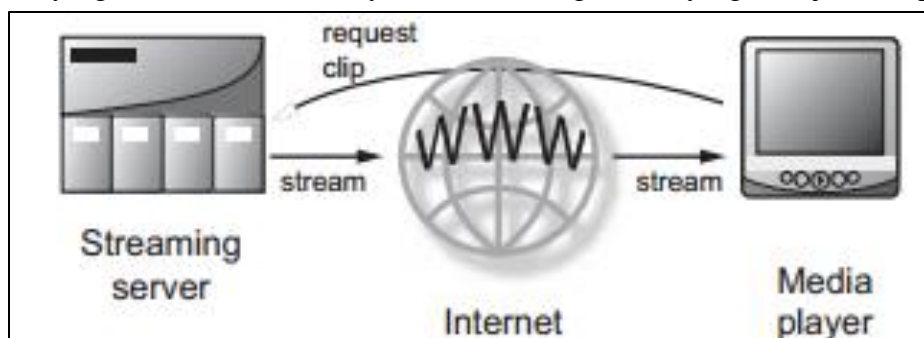
semakin halus perpindahan *frame* yang ditampilkan suatu video. Gambar 2.8 akan menunjukkan perbedaan penggunaan *frame rate*.



Gambar 2.8 Perbandingan 60 fps dan 24 fps.

2.2.3 Prinsip Kerja Video On Demand (VoD)

Layanan *Video on Demand* memiliki kemampuan untuk melakukan download dari server dan langsung ditayangkan tanpa menunggu hasil download selesai. Video yang akan ditayangkan oleh user nantinya diunduh dengan alur yang ditunjukkan pada Gambar 2.8.



Gambar 2.9 Prinsip Kerja VoD.
Sumber: Sami Saleh (2013)

Menurut Gambar 2.8, video yang akan dijadikan konten pada layanan VoD direkam kemudian disimpan pada *server* konten. Server konten ini berguna untuk menampung seluruh video yang nantinya akan ditayangkan dan diunggah ke *Streaming Server*. *Streaming Server* sendiri merupakan sebuah *web server* atau aplikasi yang telah terinstal pada server. Pada *Streaming Server*, file *audio* maupun *video* akan dilakukan transfer data secara *real time (stream)* menggunakan protokol RTPS menuju *user* melalui internet setelah *user* melakukan *request* terlebih dahulu.

2.2.4 Format Video

Format video terdiri atas *containers* dan *codec*. *Containers* adalah struktur dari *file*, dimana berbagai macam potongan file disimpan ataupun disisipkan di *codec* kemudian membungkus *video* dan komponen lainnya (*audio/metadata*) menjadi *file* beresistensi seperti .AVI, .MP4 dan lain-lainnya.

Tabel 2. 2
Aplikasi *Video Codec*

Format Kompresi	Bandwidth (bit/s)	Resolusi	Aplikasi
H.261	384 k-2 M	176 x 144 352 x 288	<i>Video conference</i>
H.263	28.8 k – 768 k	128 x 96 720 x 480	<i>Video conference</i>
MPEG-1	400 k – 1.5 M	352 x 288	CD-ROM
MPEG-2	1.5 M – 15 M	720 x 480	Siaran Televisi, DVD
MPEG-4	28.8 k – 500 k	176 x 144	<i>Fixed dan mobile web</i>
AVC, H-264	100 k – 500 k	144p, 240p, 360p, 480p, 720p, 1080p	<i>General Purpose</i>

Sumber: David Austerberry (2005)

2.2.5 *Codec*

Codec adalah algoritma yang digunakan dalam proses kompresi analog ke dalam bentuk paket-paket data. *Codec* bertujuan untuk mengurangi penggunaan *bandwidth* di dalam transfer data. *Codec* memiliki beberapa fungsi yaitu *codec* sebagai *Coder-decoder*, yang berfungsi untuk melakukan konversi data analog ke digital atau dari digital ke analog, dan *codec* sebagai *Compressor-decompressor*, yang digunakan untuk melakukan kompresi atau dekompresi terhadap audio maupun video.

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

Jenis *Codec* yang digunakan pada layanan *video on Demand* adalah AAC untuk *audio* dan H.264 untuk *video*, beberapa macam teknik *coding* lain yang sering digunakan adalah sebagai berikut seperti pada Tabel 2.3

Tabel 2.3
Audio dan video Codec

Audio CODEC	Bit rate
G.711	64 Kbps
G.726	32 Kbps
G.723.1	6,3 Kbps
G.728	16 Kbps
AAC	168 Kbps
Video CODEC	Bit rate
H.261	64 Kbps
H.263	384 Kbps
H.264/AVC	384 Kbps

Sumber: Cisco System (2006)

2.2.6 Protokol Video Streaming

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

1. User Data Protocol (UDP)

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

2. Transmission Control Protocol (TCP)

Protokol yang secara umum digunakan pada teknologi transfer data. Protokol ini digunakan untuk mengatur transmisi pada sebuah jaringan.

2. Real Time Transport Protocol (RTP)

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

4. Internet Protocol (IP)

IP terdiri dari informasi pengalamatan dan beberapa kontrol yang memungkinkan suatu data agar dapat dirutekan.

2.3 Parameter Pengaplikasian Jaringan VoD

Terdapat beberapa parameter yang dapat digunakan untuk mengukur kualitas VoD pada WiMAX 802.16d, diantaranya *Delay* (tundaan), *Packet loss*, dan *Throughput*. Subbab berikut adalah penjelasan untuk masing-masing parameter tersebut.

2.3.1 Payload Paket Data

Payload merupakan ukuran dari suatu paket data yang akan dikirimkan. *Payload* yang akan dikirim berupa *payload* paket data *video* dan *audio*. Besarnya *Payload* yang dikirimkan untuk layanan *Video on Demand* dibedakan atas paket *audio* dan paket *video*. Kedua paket tersebut memiliki nilai *payload* yang berbeda pula dan dapat dihitung menggunakan persamaan (2-1) dan (2-2) sebagai berikut:

$$P_A = B_{codec} \times frame\ rate \dots\dots\dots(2-1)$$

$$P_V = B_{codec} \times frame\ rate \dots\dots\dots(2-2)$$

dengan:

$$P_A = \text{Payload audio (byte)}$$

$$P_V = \text{Payload video (byte)}$$

$$B_{codec} = \text{Bandwidth codec (Kbps)}$$

$$frame\ rate = \text{Waktu pembentukan tiap frame (s)}$$

Sehingga jumlah paket data *audio* maupun *video* yang dienkode dan dipengaruhi oleh *Payload* masing-masing dapat dihitung dengan persamaan (2-3) dan (2-4):

$$P_A = \frac{P_{LA}}{P_{LA\ max}} \dots\dots\dots(2-3)$$

$$P_V = \frac{P_{LV}}{P_{LV\ max}} \dots\dots\dots(2-4)$$

$$P_v = \text{Jumlah paket video}$$

$$P_a = \text{Jumlah paket audio}$$

$$P_{LV\ max} = \text{Payload video maksimum berdasarkan codec (bit)}$$

$$P_{LA\ max} = \text{Payload audio maksimum berdasarkan codec (bit)}$$

Jumlah paket data *Video on Demand* yang ditambahkan dengan *header* NALU, RTP, UDP, dan IP secara matematis dapat dihitung menggunakan persamaan:

$$P_{audio-enc} = P_{LA} + (P_A \times (NALU + H_{RTP} + H_{UDP} + H_{IP})) \dots\dots\dots(2-5)$$

$$P_{video-enc} = P_{LV} + (P_V \times (NALU + H_{RTP} + H_{UDP} + H_{IP})) \dots\dots\dots(2-6)$$

Panjang paket data *Video on Demand* dituliskan pada persamaan berikut:

$$P_{data} = P_{audio-enc} + P_{video-enc} \dots\dots\dots(2-7)$$

dengan:

$$P_{data} = \text{panjang paket data Video on Demand (byte)}$$

$$P_{audio-enc} = \text{panjang paket audio (byte)}$$

$$P_{video-enc} = \text{panjang paket video (byte)}$$

2.3.2 Bit rate Video

Bit rate Video merupakan ukuran berapa banyak bit yang dapat ditransmisikan per detik untuk *bandwidth* yang disediakan. *Bitrate Video* untuk video dengan format AVC/H.264 ditunjukkan oleh persamaan berikut (Kush Amerasinghe, 2009):

$$\lambda_{video} = W_{video} \times H_{video} \times FR \times K_{kompresi} \times MR \times CD \dots \dots \dots (2-8)$$

dimana,

$$K_{kompresi} = \frac{C_{video}}{W_{video} \times H_{video} \times FR \times CD} \dots \dots \dots (2-9)$$

dengan:

λ_{video} = *Bit rate Video* (bps)

W_{video} = Lebar *video*(bit)

H_{video} = Tinggi *video*(bit)

FR = *Frame rate*, jumlah *frame* dalam *video* setiap detiknya

MR = *Motion Rank*, bernilai 1 hingga 4 tergantung gerakan yang ada pada *video*

$K_{kompresi}$ = Konstanta kompresi *Codec* AVC/H.264

C_{video} = *bitrate codec video* (bps)

CD = *Color Depth* (32 bit)

Jumlah gerakan dalam *video* dikategorikan menjadi tiga urutan yang disebut *motion rank*. Tiga jenis tersebut adalah *low motion*, *medium motion* dan *high motion*.

1. *Low motion* adalah *video* yang memiliki gerakan yang tidak terlalu banyak seperti saat melakukan wawancara atau melakukan *review* suatu produk. Nilai dari *low motion* = 1.
2. *Medium motion* adalah *video* yang memiliki gerakan-gerakan yang dinamis dengan beberapa perubahan *scene* atau adanya perubahan *background* secara cukup drastis. Contoh nya adalah *VLog* dan Olahraga. Nilai dari *medium motion* = 1,5 - 3.
3. *High motion* adalah *video* yang memiliki gerakan yang sangat dinamis seperti *action movies* dengan pergerakan dan perubahan *scene* yang sangat cepat. Nilai dari *high motion* = 4.

2.3.3 Delay End-to-End

Delay merupakan waktu yang dibutuhkan untuk sebuah paket data dikirimkan dari satu titik ke titik lain yang menjadi tujuannya. *Delay End to End* merupakan *delay* total keseluruhan *delay* yang diperlukan untuk mengirimkan data dari sisi *server* ke *subscriber*

lain pada jaringan *fixed* WiMAX. *Delay End to End* dapat dihitung dengan menggunakan persamaan berikut:

$$t_{End\ to\ End} = t_{proses} + t_{prop} + t_{trans} + t_w \dots \dots \dots (2-10)$$

dengan:

$$t_{End\ to\ End} = \text{delay total pada MAN (s)}$$

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

$$t_{prop} = \text{delay propagasi (s)}$$

$$t_{trans} = \text{delay transmisi (s)}$$

$$t_w = \text{delay antrian (s)}$$

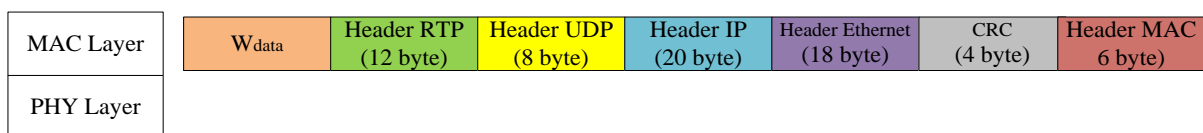
2.3.3.1 Delay Proses

Delay proses merupakan penjumlahan dari *delay* enkapsulasi dan *delay* dekapsulasi. *Delay* enkapsulasi merupakan waktu yang dibutuhkan suatu paket untuk menambahkan *header* ketika paket data hendak ditransmisikan dari *Server*. Sedangkan *delay* dekapsulasi merupakan proses melepaskan seluruh *header* ketika paket sudah sampai *User*.

Header yang digunakan untuk transmisi melalui jaringan *WiMAX* antara lain sebagai berikut :

1. Ukuran *header* IPv4 (L_{IPv4}) = 20 byte/paket
2. Ukuran *header* NALU (L_{NALU}) = 1 byte/paket
3. Ukuran *header* UDP (L_{UDP}) = 8 byte/paket
4. Ukuran *header* RTP (L_{RTP}) = 12 byte/paket
5. Ukuran *header* Ethernet ($L_{Ethernet}$) = 14 byte/paket
6. Ukuran *header* CRC (L_{CRC}) = 4 byte/paket
7. Ukuran *header* MAC *WiMAX* (L_{MAC}) = 6 byte/paket

WIMAX LAYER



Gambar 2.10 Format paket data MAC WiMAX.

Sumber: Nuaymi (2007)

Berikut ini adalah persamaan untuk menghitung *delay* enkapsulasi dan dekapsulasi :

$$L_{tot} = L_{IPv4} + L_{NALU} + L_{UDP} + L_{RTP} + L_{Ethernet} + L_{MAC} + L_{CRC} \dots \dots \dots (2-11)$$

$$C_{proses} = N_k \times 8 \times FR \dots \dots \dots (2-12)$$

$$t_{enc/dec} = \frac{L_{tot}}{C_{proses}} \dots\dots\dots(2-13)$$

dengan

L_{tot} = Panjang *Header* total (bit)

$t_{enc/dec}$ = *delay* enkapsulasi dan *delay* dekapulasi

C_{proses} = kecepatan pemrosesan pada terminal pengirim (bps);

N_k = jumlah total data yang dikirimkan (bit);

sehingga *delay* proses dapat dirumuskan sebagai berikut:

$$t_{proc} = t_{enc} + t_{dec} \dots\dots\dots(2-14)$$

dengan:

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

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

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

2.3.3.2 Delay Propagasi

Delay propagasi adalah waktu yang dibutuhkan untuk merambatkan paket data melalui media transmisi. *Delay* propagasi gelombang radio dapat ditulis dengan persamaan:

$$t_{prop} = \frac{d}{v} \dots\dots\dots(2-15)$$

dengan:

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

d_{max} = jarak jangkauan *base station* dan *subscriber station* (m)

v = kecepatan sinyal pada media *wireless* (m/s)

2.3.3.3 Delay Transmisi

Delay transmisi adalah waktu yang dibutuhkan sebuah paket untuk diletakkan ke media transmisi menggunakan media kabel *Ethernet* lalu beralih ke media *WiMAX*. Sehingga *delay* yang diperoleh adalah penjumlahan *delay* saat paket ditransmisikan melalui kabel *Ethernet* dan *WiMAX*. *Delay* transmisi dapat ditentukan dengan persamaan (Mischa Schwartz,1987)

$$t_{trans} = \frac{W_{Packet}}{\lambda} \times 8 \dots\dots\dots(2-16)$$

dengan:

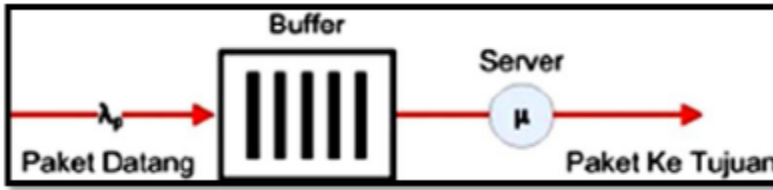
t_{trans} = *delay* transmisi (s)

W_{packet} = panjang total *frame* yang dikirimkan (byte)

λ = *Throughput* (bps)

2.3.3.4 Delay Antrian

Delay antrian adalah waktu tunggu yang diperlukan suatu paket data untuk diproses pada suatu perangkat. Delay antrian model M/M/1 dapat dihitung dengan persamaan:



Gambar 2.11 Model antrian M/M/1

$$\mu = \frac{C_{WiMAX}}{W_{paket}} \dots \dots \dots (2-17)$$

$$\lambda_w = \frac{N}{\tau} \dots \dots \dots (2-18)$$

$$t_w = \frac{\lambda_w}{\mu(\mu - \lambda_w)} \dots \dots \dots (2-19)$$

dengan :

- t_w = delay antrian (s) N = rata-rata paket terkirim
- λ_w = kecepatan kedatangan (paket/s) τ = rata-rata waktu pengiriman
- μ = waktu pelayanan (paket/s)
- C_{WiMAX} = kapasitas kanal (bps)
- W_{paket} = Ukuran Paket Data (bit)

2.3.4 Probabilitas Packet Loss

Probabilitas Packet Loss merupakan besarnya kemungkinan jumlah paket data yang hilang pada saat transmisi. Probabilitas packet loss total pada WiMAX ditentukan berdasarkan packet loss pada jaringan yang berbasis protokol UDP/RTP/IP seperti ditunjukkan pada persamaan (Pritchard. Et.al, 1993)

$$\rho_{tot} = 1 - [(1 - \rho_{net})(1 - \rho_{payload})] \dots \dots \dots (2.20)$$

dengan:

- ρ_{tot} = Probabilitas packet loss total
- ρ_{net} = Probabilitas packet loss pada jaringan WiMAX
- $\rho_{payload}$ = Probabilitas packet loss untuk Video on Demand

Besarnya probabilitas packet loss dalam persen dirumuskan dalam persamaan :

$$\rho_{tot}(\%) = \rho_{tot} \times 100 \% \dots \dots \dots (2.21)$$

a. Probabilitas *Packet Loss* pada Server

Packet Loss yang terjadi pada layanan *video on demand* dihitung melalui persamaan :

$$\rho_{\text{payload}} = P_{\text{size}} \cdot \rho_b \dots\dots\dots(2.22)$$

dengan:

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

ρ_b = BER (10^{-7})

b. Probabilitas *Packet Loss* pada Jaringan

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

$$P_{\text{net}} = \frac{3}{2k} \operatorname{erfc} \left(\sqrt{\frac{k \cdot E_b}{10 N_o}} \right) \dots\dots\dots(2-23)$$

Keterangan:

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

P_{net} = *Probabilitas bit salah/bit error rate* pada modulasi 64 QAM

c. Signal to Noise Ratio (SNR)

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

$$\text{SNR} = 10 \operatorname{Log} \frac{P_r}{N_t} \dots\dots\dots(2-24)$$

dengan:

SNR = *signal to noise ratio* (dB)

P_r = daya yang diterima oleh penerima (mW)

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

Dengan daya noise pada saluran transmisi dinyatakan dalam persamaan berikut :

$$N_t = 10 \operatorname{log} kT + 10 \operatorname{log} B + \text{NF} \dots\dots\dots(2-25)$$

dengan :

N_t = daya *noise* saluran transmisi (dB)

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

T = suhu *absolute* (300° K)

NF = *noise figure*

B = *bandwidth* (Hz)

d. Perhitungan *bit rate* 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):

$$R = 2B \times \log_2 n \dots\dots\dots (2-26)$$

dengan:

n = banyaknya bit pada suatu jenis modulasi

B = *bandwidth* jaringan yang dipergunakan (Hz)

R = *bit rate* modulasi (bps)

e. Energy Bit-to-Noise Ratio

E_b/N_0 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_0 ini dipengaruhi oleh SNR, dan *bit rate* pada modulasi.

Perhitungan E_b/N_0 , 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):

$$\frac{E_b}{N_0} = \text{SNR} \times \frac{B}{R} \dots\dots\dots (2-27)$$

dengan:

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

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

B = *bandwidth* sistem dalam dB

R = *bit rate* modulasi dalam dB

2.3.5 Throughput

Throughput merupakan 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 2-27 (Stallings, 2005).

$$\lambda = \frac{(1-\rho)}{t_{trans}[1+(\alpha-1)\rho]} \dots\dots\dots (2-27)$$

Keterangan :

λ = *throughput* (paket/s)

ρ = *probabilitas packet loss*

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

Simbol α merupakan konstanta *propagasi* dengan persamaan 2-28.

$$\alpha = \left(3 + \frac{2t_{prop}}{t_{paketisasi}} \right) \dots\dots\dots (2-28)$$

Keterangan :

α = konstanta *propagasi*

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

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

2.4 Rekomendasi dan Perangkat Pendukung

2.4.1 Rekomendasi ITU-T G.1010

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

Tabel 2.4

Standar *Packet Loss* ITU-T G.1010

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

Sumber: ITU-T G.1010 (2002)

2.4.2 Rekomendasi ITU-T G.114

Berdasarkan ITU-T G.114, memiliki persyaratan delay pada *transmission system and media, digital system and networks* seperti yang ditunjukkan pada Tabel 2.5.

Tabel 2.5

Standar *Delay* ITU-T G.114

<i>Delay</i> (s)	Kualitas
$t_{end\ to\ end} < 0.15$	Baik
$0.15 \leq t_{end\ to\ end} \leq 0.4$	Cukup, masih dapat diterima
$t_{end\ to\ end} > 0.4$	Buruk

Sumber: ITU-T G.114 (2011)

2.4.3 Wireshark

Wireshark merupakan perangkat lunak berbasis *open-source* dan gratis. Wireshark digunakan untuk menyelesaikan permasalahan pada jaringan, analisis, perangkat lunak dan pengembangan protokol komunikasi.



Gambar 2.12 Logo Wireshark.

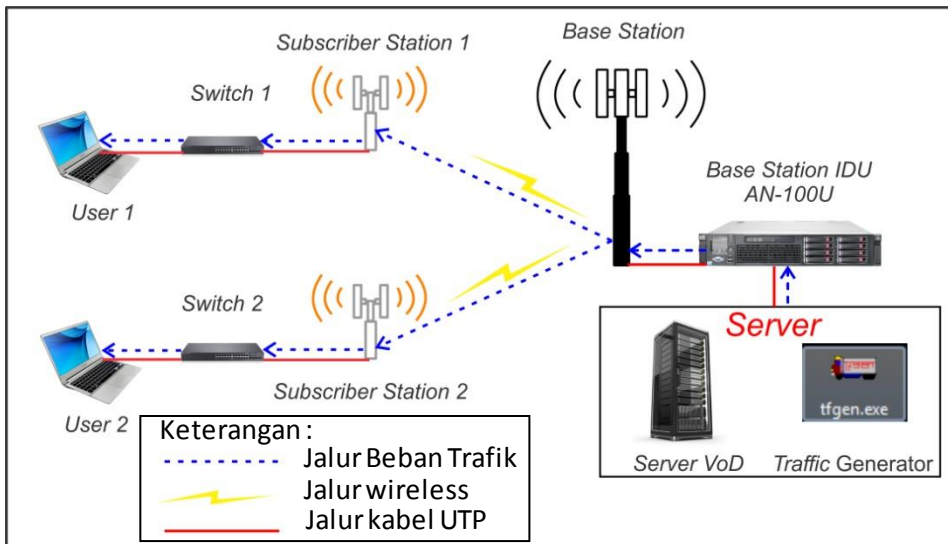
Sumber: <http://www.wireshark.org>

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

1. *Capture*. Wireshark menangkap paket data yang melewati adapter kabel jaringan maupun wireless pada pengguna.
2. *Decode*. *Decode* atau dekode adalah proses mengubah suatu protokol menjadi sebuah informasi yang dapat diketahui.
3. *Analyze*. *Analyze* dapat menampilkan beberapa parameter terkait performansi maupun karakteristik paket data.

2.4.4 Traffic Generator (TfGen)

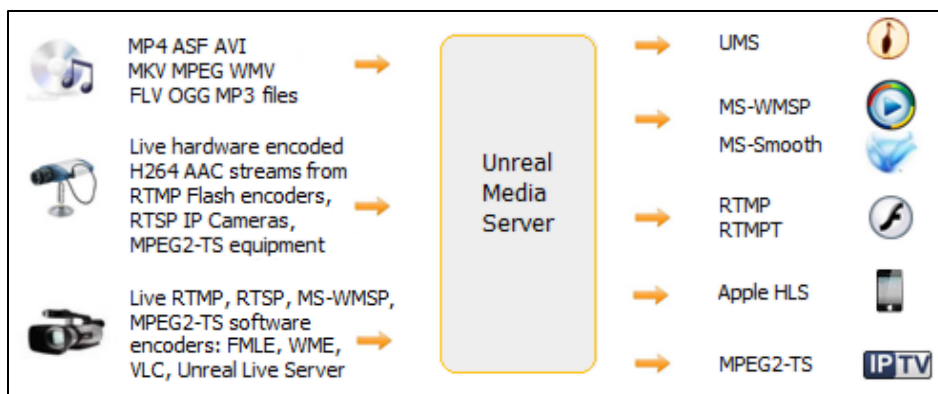
Traffic Generator merupakan perangkat lunak yang berfungsi untuk mengirimkan satu paket atau banyak paket-paket dalam jaringan. TfGen dipasang pada sisi server dan diatur agar mengirimkan paket-paket ke semua user melalui jaringan Testbed WiMAX, sehingga jaringan uji dibuat seolah-olah seperti jaringan sebenarnya yang memiliki trafik data.



Gambar 2.13 Pembebanan Trafik

2.4.5 Unreal Media Server dan Streaming 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 2.14 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 (unmediaserver.net). Sedangkan

Streaming Media Player merupakan aplikasi yang dapat memutar video yang telah disimpan dalam *Unreal Media Server* secara *streaming*.