

BAB II

VIDEO STREAMING PADA WIMAX 802.16d

2.1 *Worldwide Interoperability for Microwave Access (WiMAX)*

Worldwide Interoperability for Microwave (WiMAX) merupakan standar *Broadband Wireless Access (BWA)* dengan kemampuan untuk menyalurkan data kecepatan tinggi (layaknya teknologi xDSL pada jaringan *wireline*) [3]. Banyak kemampuan lebih yang ditawarkan oleh teknologi WiMAX dibanding teknologi sebelumnya seperti kemampuan diterapkan dalam kondisi *Non Line of Sight (NLOS)*, aplikasinya baik untuk *fixed, nomadic, portable* maupun *mobile* [3].

2.1.1 Standar WiMAX

WiMAX merupakan standar internasional tentang BWA yang mengacu pada standar *Institute of Electrical and Electronics Engineering (IEEE) 802.16* [3]. Standar ini kemudian dikembangkan lebih lanjut oleh forum gabungan antar perusahaan-perusahaan dunia terkait (produsen produk-produk *wireless*, produsen-produsen chip, operator-operator *wireless*), atau disebut dengan WiMAX FORUM [3]. Forum ini adalah sertifikasi dari kemampuan *interoperability* perangkat-perangkat BWA yang akan diproduksi. Secara sederhana perkembangan standar 802.16 dapat diuraikan sebagai berikut [3] :

- **802.16**

Standar ini mengatur pemanfaatan di *band* frekuensi 10–66GHz. Aplikasi yang mampu didukung baru sebatas dalam kondisi *Line of Sight (LOS)*.

- **802.16a**

Menggunakan frekuensi 2–11GHz, dapat digunakan untuk lingkungan *Non Line of Sight*. Standar ini difinalisasi pada Januari 2003. Terdapat 3 spesifikasi pada *physical layer* di dalam 802.16a, yaitu :

- *Wireless MAN-SC*: menggunakan format modulasi *single carrier*.
- *Wireless MAN-OFDM* : menggunakan *Orthogonal Frequency Division Multiplexing (OFDM)* dengan *256 point Fast Fourier*

Transform (FFT). Modulasi ini bersifat *mandatory* untuk *non licensed band*.

- Wireless MAN-OFDMA : menggunakan *Orthogonal Frequency Division Multiple Access* (OFDMA) dengan 2048 point FFT.

- **802.16d**

Merupakan standar yang berbasis 802.16 dan 802.16a dengan beberapa perbaikan. 802.16d juga dikenal sebagai 802.16-2004. Frekuensi yang digunakan sampai 11 GHz. Standar ini telah difinalisasi pada 24 Juni 2004. Terdapat dua opsi dalam transmisi pada 802.16d yaitu *Time Division Duplex* (TDD) maupun *Frequency Division Duplex* (FDD).

- **802.16e**

Standar ini memenuhi kapabilitas untuk aplikasi *portability* dan *mobility*. Standar ini telah difinalisasi di akhir tahun 2005. Berbeda dengan standar sebelumnya, maka antara standar 802.16d dan 802.16e tidak bisa dilakukan *interoperability* sehingga diperlukan *hardware* tambahan bila akan mengoperasikan 802.16e.

Sampai saat ini, Standar WiMAX yang dikenal adalah 2 tipe standar yaitu 802.16d (802.16-2004) untuk aplikasi *fixed* dan *nomadic* dan standar 802.16e (802.16-2005) untuk aplikasi *portable* dan *mobile* [3].

2.1.2 Aliran Trafik Pada 802.16d

Secara umum, sistem WiMAX terdiri dari *Base Station* (BS), *Subscriber Station* (SS) dan server di belakang BS seperti *Network Management System* (NMS) serta koneksi ke jaringan. Konfigurasi WiMAX dibagi menjadi 3 bagian yaitu SS, BS dan *transport site*. SS terletak di lingkungan pelanggan, pada 802.16d lokasinya *Fixed*, sedangkan BS biasanya satu lokasi dengan jaringan operator (PSTN/Internet).

Base Station (BS) merupakan perangkat *transceiver* (*transmitter* dan *receiver*) yang biasanya dipasang satu lokasi (*colocated*) dengan jaringan *internet protocol*. Dari BS ini akan disambungkan ke beberapa SS yang berupa *Customer Premise Equipment* (CPE) dengan *media interface* gelombang radio (RF) yang mengikuti standar WiMAX. Antena yang dipakai di BS dapat berupa antena sektoral 60°,

90°, atau 120° tergantung dari area yang akan dilayani. *Remote Stations* atau CPE terdiri dari *Outdoor Unit* (ODU) dan *Indoor Unit* (IDU), dimana perangkat radionya ada yang terpisah dan ada yang terintegrasi dengan antena.

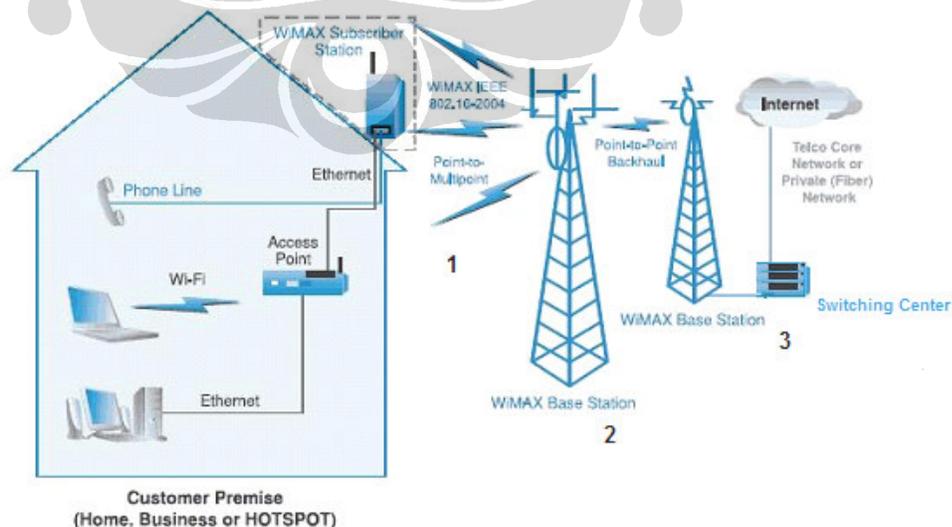
Teknologi WiMAX 802.16d dapat mencakup area sekitar 50 kilometer, dimana kepada ratusan pengguna akan di-*share* sinyal dan kanal untuk mentransmisikan data dengan kecepatan sampai 70 Mbps pada 20 MHz [4].

Aspek keamanan merupakan aspek yang sangat penting. Sistem pengamanan data WiMAX 802.16d dilakukan pada *physical layer* (PHY) dan *data link layer Medium Access Control* (MAC) dalam suatu arsitektur jaringan, tepatnya pada *Base Station* (BS) untuk didistribusikan ke wilayah sekelilingnya dan *Subscriber Station* (SS), untuk komunikasi *Point to Point* (P2P) [4]. BS dihubungkan secara langsung dengan jaringan umum (*public network*).

Aliran trafik pada WiMAX terdiri atas tiga bagian, yaitu [4]:

1. Pelanggan mengirimkan data dengan kecepatan hingga 75 Mbps dari SS ke BS.
2. BS akan menerima sinyal dari berbagai pelanggan dan mengirimkan pesan melalui *wireless* atau kabel ke *switching center* melalui protokol IEEE 802.16.
3. *Switching center* akan mengirimkan pesan ke *internet service provider* (ISP) atau *public switched telephone network* (PSTN).

Aliran trafik pada 802.16d digambarkan pada Gambar 2.1



Gambar 2.1 Aliran Trafik pada Wimax 802.16d

Teknologi yang digunakan untuk komunikasi antara SS dan BS adalah teknologi TDMA. Untuk menjamin kerahasiaan data pada pengguna, maka pengiriman atau penerimaan data dari SS dan BS dienkripsi menggunakan X.509 yang disertifikasi oleh RSA [4]. Ancaman yang umum pada pengguna menggunakan teknologi WiMAX adalah:

- Pencurian sinyal atau layanan
- Pencurian data user
- *Cloning*

Pada standar IEEE 802.16 digunakan metode untuk meningkatkan keamanan berupa otentikasi, otorisasi dan enkripsi [4]. Otentikasi yang digunakan pada SS adalah X.509 dengan RSA *Public Key Cryptography Standard (PKCS)*.

Otentikasi dan otorisasi pada SS menggunakan X.509 dengan kunci publik digunakan untuk mengidentifikasi informasi, misalnya *UserID*, *SS name* dan lain sebagainya [4]. Informasi ini akan terus diidentifikasi selama komunikasi antara SS dan BS masih berlangsung.

Enkripsi yang digunakan dalam standar IEEE 802.16 adalah 56-bit DES pada mode *cyclic block chaining (CBC)*, di mana kesalahan yang terjadi pada *cyphertext* tidak dipropagasikan ke dalam *plaintext* dengan menerapkan algoritma *multiple encryption* [4].

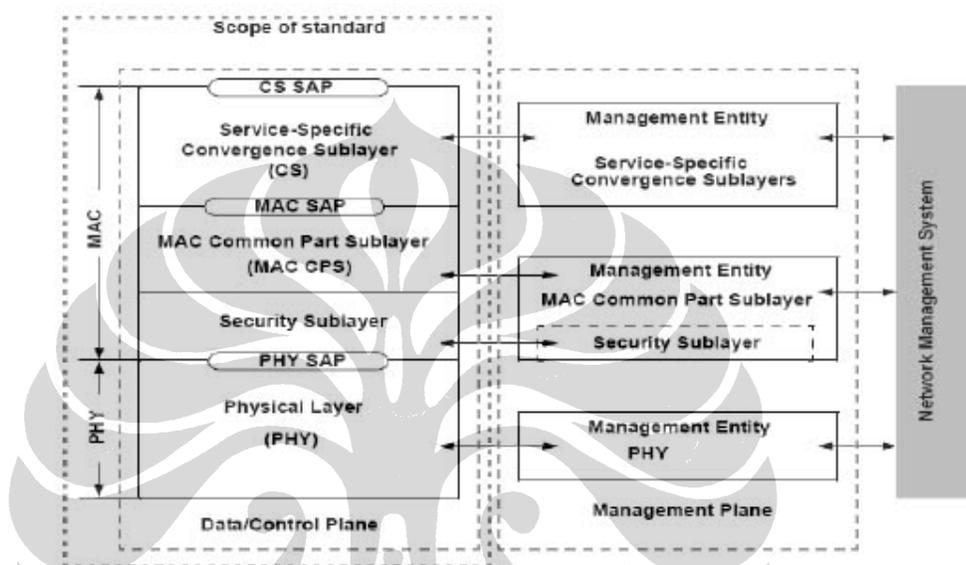
2.1.3 Otentikasi dan Registrasi Pada *Subscriber Station (SS)*

Setiap SS memiliki dua buah sertifikasi yaitu X.509 dan sertifikasi dari perusahaan [4]. Sertifikasi yang menghubungkan antara 48-bit MAC SS dan kunci RSA dikirimkan dari BS ke SS dalam bentuk *Authorization Request (AR)* dan *Authentication Information (AI)* [4]. Setelah berhasil melakukan proses otentikasi dan otorisasi maka SS akan tercatat dalam jaringan dan *subscriber* akan menerima sebuah IP address dari server DHCP dan dapat mengakses WiMAX.

2.1.4 Struktur Layer

Karakteristik standar 802.16 ditentukan oleh spesifikasi teknis dari *Physical (PHY) Layer* dan *Medium Access Control (MAC) Layer* [4]. Perbedaan

karakteristik kedua layer ini akan membedakan varian-varianya. Pada Gambar 2.2 ditunjukkan lingkup dari standar yang meliputi PHY dan MAC. Sedangkan *Network Management System* (NMS) dan *Management Plane* dapat berbeda-beda mengikuti strategi disain dari masing-masing manufaktur atau *vendor* pembuatnya [4].



Gambar 2.2 Layer PHY dan MAC pada standar 802.16

Physical layer menjalankan fungsi mengalirkan data di level fisik. *MAC Layer* berfungsi sebagai penterjemah protokol-protokol yang ada di atasnya seperti ATM dan IP. *MAC layer* dibagi lagi menjadi tiga sub-layer [4]: *Service-Specific Convergence Sublayer (SS-CS)*, *MAC Common Part Sublayer*, dan *Security Sublayer*.

2.1.4.1 PHY Layer

Pada standar WiMAX, fungsi-fungsi penting yang di atur pada PHY adalah: OFDM, *Duplex Sistem*, *Adaptive Modulation*, *Variable Error Correction*, dan *Adaptive Antenna System (AAS)*. Semua fungsi-fungsi ini secara bersama-sama memberikan keunggulan yang cukup berarti dibandingkan dengan BWA yang ada sebelumnya.

Dengan teknologi OFDM memungkinkan komunikasi berlangsung dalam kondisi *multipath* LOS dan NLOS antara *Base Station* (BS) dan *Subscriber Station* (SS). Metode OFDM yang digunakan untuk WiMAX adalah *Fast Fourier Transfer* (FFT) 256 [4]. Fitur PHY untuk sistem *duplex* pada standar WiMAX bisa diterapkan pada *Frequency Division Duplexing* (FDD), *Time Division Duplexing* (TDD) atau keduanya TDD dan FDD. Fitur ini memberikan kemudahan pengaturan spektrum frekuensi yang akan digunakan oleh para operator agar didapatkan efisiensi spektrum yang optimal. Hal ini juga sejalan dengan penggunaan kanal (kanalisasi) yang diperkenankan, yaitu dari 1.7 MHz sampai dengan 20 MHz [4].

Varian PHY yang diadopsi dari standar 802.16 adalah WirelessMAN-OFDM dan WirelessMAN-OFDMA untuk *licensed frequency* serta Wireless HUMAN untuk frekuensi *Unlicensed National Information Infrastructure* (UNII) dan frekuensi *unlicensed* lainnya [4].

2.1.4.2 MAC Layer

WiMAX MAC protokol didesain untuk aplikasi *Point to Multi Point* (PMP) [4]. Berbeda dengan WiFi, mekanisme pengalokasian dipersiapkan untuk menangani ratusan terminal per kanal, dan setiap terminal memungkinkan lagi untuk penggunaan secara bersama dengan beberapa pengguna akhir.

Pada *MAC Layer* digunakan dua jalur data berkecepatan data tinggi untuk komunikasi dua arah antara BS dan SS, masing-masing disebut dengan *Up Link* (UL) untuk komunikasi menuju ke BS, dan *Down Link* (DL) untuk komunikasi dari BS [4]. Secara umum DL ditransmisikan secara *broadcast* dari BS dan semua SS menerima sinyal DL tersebut tanpa koordinasi langsung antar SS yang ada. Pada penggunaan sistem TDD, ditentukan periode transmit untuk DL dan UL.

MAC Layer mempunyai karakteristik *connection-oriented* dan setiap sambungan diidentifikasi oleh 16-bit *Connection Identifiers* (CID) [4]. CID digunakan untuk membedakan kanal UL dan lainnya. Setiap SS memiliki *MAC Address* dengan lebar standar 48 bit. Dalam mekanisme sambungan antar SS dan BS, terdapat tiga jenis koneksi manajemen untuk setiap arah, yang masing-masing

memerlukan tingkat penanganan QoS yang berbeda [4]. Ketiga sambungan tersebut adalah :

- *Basic Connection*, menjalankan transfer yang relatif singkat, melibatkan *Radio Link Control* (RLC), dan kritis terhadap waktu
- *Primary Management Connection*, menjalankan transfer relatif lama, lebih toleran terhadap *delay*, digunakan untuk proses otentikasi dan *connection setup*.
- *Transport Connection*, digunakan untuk pengaturan layanan, QoS dan parameter-parameter trafik.

2.1.5 Sublayer Privasi

Sublayer privasi menyediakan sistem pengamanan data dengan enkripsi diantara BS dan SS. BS memproteksi pengaksesan data dengan cara enkripsi pada seluruh jaringan. *Sublayer* privasi dibedakan menjadi dua protokol [4], yaitu:

- a. Protokol *enkapsulasi* yang akan bertanggung jawab terhadap data yang melewati jaringan BWA.
- b. Protokol *Key management* (*Privacy Key Management* (PKM)) yang menyediakan keamanan distribusi antara BS dan SS.

2.2 Quality of Service (QoS)

Pada jaringan, *Quality of Service* (QoS) mengacu pada kemampuan untuk menyediakan perlakuan yang berbeda untuk kelas yang berlainan dari trafik. Tujuan utamanya adalah untuk meningkatkan kegunaan secara keseluruhan dari jaringan dengan mengizinkan ke nilai yang tertinggi atau yang memiliki tingkat sensitifitas yang tinggi pada *flow*.

Pada WiMAX, fungsi pengaturan QoS dijalankan oleh *Medium Access Control* (MAC) [5]. QoS menggunakan arsitektur MAC *connection-oriented*. Dimana semua koneksi *downlink* dan *uplink* dikendalikan oleh *Base Station* (BS) yang melayaninya. Sebelum transmisi data dilangsungkan, BS dan SS membentuk *link logical* satu arah untuk koneksi antara MAC-layer. Setiap koneksi

diidentifikasi dengan sebuah *connection identifier* (CID) [5], yang menjadi alamat sementara untuk transmisi data pada link tertentu.

Dalam WiMAX juga terdapat konsep *service flow*. *Service flow* adalah aliran satu arah dimana paket memiliki susunan parameter QoS tertentu, dan diidentifikasi dengan sebuah *service flow identifier* (SFID) [5]. Parameter QoS dapat terdiri dari prioritas trafik, *sustained traffic rate* maksimum, *burst rate* maksimum, *tolerable rate* minimum, tipe *scheduling*, tipe *ARQ*, *delay* maksimum, *tolerated jitter*, *service data unit type* dan ukuran, mekanisme permintaan *bandwidth* yang digunakan [5], dan seterusnya.

Service flow dapat ditentukan melalui sistem manajemen jaringan yang dibuat secara dinamis dengan mendefinisikan mekanisme pensinyalan dalam standar. *Base station* bertanggungjawab untuk menerbitkan SFID dan memetakannya ke CID unik. *Service flow* juga dapat dipetakan ke titik kode *Differentiated Services* (*DiffServ*) atau label aliran *Multi Protocol Label Swiching* (MPLS) untuk mengizinkan QoS berbasis *end-to-end* [5].

Untuk mendukung berbagai jenis aplikasi, WiMAX mendefinisikan lima layanan penjadwalan yang dapat dilihat pada Tabel 2.1, layanan penjadwalan harus didukung oleh penjadwal MAC *base station* untuk memindahkan data melalui koneksi [5], yaitu :

1. *Unsolicited Grant Service* (UGS)

UGS digunakan untuk layanan yang membutuhkan jaminan transfer data dengan prioritas paling utama. Dengan demikian layanan dengan kriteria UGS ini memiliki karakteristik :

- Seperti halnya layanan *Constant Bit Rate* (CBR) pada *Asynchronous Transfer Mode* (ATM), yang dapat memberikan transfer data secara periodik dalam ukuran yang sama (*burst*).
- Untuk layanan-layanan yang membutuhkan jaminan *real-time*.
- Efektif untuk layanan yang sensitif terhadap *throughput*, *latency* dan *jitter* seperti layanan pada *Time Division Multiplexing* (TDM).
- *Bandwidth* maksimum dan minimum yang ditawarkan sama.

2. Real Time Polling Service (rtPS)

- Efektif untuk layanan yang sensitif terhadap *throughput* dan *latency* namun dengan toleransi yang lebih longgar bila dibandingkan dengan UGS.
- Untuk ukuran paket data yang berubah secara periodik.
- Garansi *rate* dan syarat *delay* telah ditentukan.

3. Non-Real-Time Polling Service (nrtPS)

- Efektif untuk aplikasi yang membutuhkan *throughput* yang intensif dengan garansi minimal pada *latency*-nya.
- Layanan *non real-time* dengan *regular variable size burst*.
- Layanan mungkin dapat diperluas sampai *full bandwidth* namun dibatasi pada kecepatan maksimum yang telah ditentukan.
- Garansi *rate* diperlukan namun *delay* tidak digaransi.

4. Best Effort (BE)

- Untuk trafik yang tidak membutuhkan jaminan kecepatan data (*best effort*).
- Tidak ada jaminan (*requirement*) pada *rate* atau *delay*-nya.

Tabel 2.1 QoS pada WiMAX 802.16d

QoS Category	Applications	QoS Specifications
UGS Unsolicited Grant Service	VoIP	<ul style="list-style-type: none"> • Maximum Sustained Rate • Maximum Latency Tolerance • Jitter Tolerance
rtPS Real-Time Polling Service	Streaming Audio or Video	<ul style="list-style-type: none"> • Minimum Reserved Rate • Maximum Sustained Rate • Maximum Latency Tolerance • Traffic Priority
nrtPS Non-Real-Time Polling Service	File Transfer Protocol (FTP)	<ul style="list-style-type: none"> • Minimum Reserved Rate • Maximum Sustained Rate • Traffic Priority
BE Best-Effort Service	Data Transfer, Web Browsing, etc.	<ul style="list-style-type: none"> • Maximum Sustained Rate • Traffic Priority

WiMAX mendefinisikan beberapa parameter dan fitur yang memfasilitasi implementasi untuk sebuah penjadwal yang efektif [5], diantaranya :

- Mendukung definisi detail parameter untuk kebutuhan QoS, dan berbagai mekanisme pada kondisi trafik sinyal efektif, serta kebutuhan QoS detail pada *uplink*.
- Dukungan pada alokasi sumber daya dinamis tiga dimensi pada MAC *layer*. Sumber daya dapat dialokasikan dalam waktu (*time slots*), frekuensi (*subcarriers*) dan ruang (beberapa antenna) pada basis *frame-by-frame*.
- Mendukung *feedback* cepat dari informasi kualitas kanal untuk memungkinkan *scheduler* memilih *coding* yang tepat serta modulasi (*burst profile*) untuk setiap alokasi.
- Dukungan untuk permutasi *subcarrier* berkelanjutan, yang mengizinkan penjadwal untuk mengeksplorasi keragaman *multiuser* dengan mengalokasikan setiap *subscriber* pada *subchannel* terkuat yang berhubungan.

Hal yang harus diperhatikan bahwa implementasi dari sebuah penjadwal yang efektif merupakan hal yang sangat penting untuk keseluruhan kapasitas dan kinerja sistem WiMAX.

2.3 Video Streaming

Streaming sebenarnya adalah proses pengiriman data secara terus-menerus yang dilakukan secara *broadcast* melalui jaringan komputer atau jaringan internet untuk ditampilkan oleh aplikasi *streaming* pada perangkat klien [6]. Paket-paket data yang dikirimkan telah dikompresi untuk memudahkan pengirimannya. Ada dua tipe *streaming* [6], yaitu :

- *Live streaming*
Media yang di-*playback*, di-*capture* secara langsung (*live*) dari kamera atau dari stasiun radio *online*. *Live Streaming* biasanya digunakan pada tipe video yang diliput merupakan liputan langsung.
- *Real-time streaming*
Media yang di-*playback* diambil dari file video/audio yang tersimpan di server (*on-demand*). Pada beberapa server memungkinkan untuk *rewind*,

fastforward media. *Real-Time Streaming* biasanya digunakan jika mempertimbangkan manajemen *bandwidth* serta perlindungan hak cipta.

Pada penelitian ini, *streaming* yang digunakan termasuk dalam *Real-time Streaming*, karena memungkinkan untuk melakukan *rewind* atau memungkinkan simulasi secara berulang-ulang. *Video Streaming* menggunakan protokol-protokol sebagai berikut:

1. Protokol Utama

- *User Datagram Protocol* (UDP), merupakan solusi lebih mudah dan lebih efisien namun menimbulkan banyaknya *data loss*.
- *Transmission Control Protocol* (TCP), menjamin penyampaian yang tepat namun dapat terjadi *timeout* sehingga klien membutuhkan *buffer* yang cukup.

2. Protokol Transport

- *Real-Time Streaming Protocol* (RTSP), mengizinkan klien untuk mengendalikan *streaming* media server secara *remote*, mengeluarkan perintah *VCR-like* seperti “*play*” dan “*pause*” dan mengizinkan akses file berbasis waktu pada server.
- *Real-Time Transport Protocol* (RTP), mendefinisikan standarisasi format paket untuk penyampaian audio dan video di internet.
- *Real-time Transport Control Protokol* (RTCP), merupakan protokol pengendalian paket data pada RTP yang juga berguna untuk menjamin *QoS video streaming*. RTCP digunakan secara periodik untuk mentransmisikan *control packet* untuk pengemasan pada sesi *video streaming*.

Kualitas audio/video yang diterima tergantung *bandwidth* klien. Jika *bitrate download* klien berbeda jauh dengan *bitrate playback* maka media tidak akan tampil. *Streaming* disarankan untuk digunakan pada siaran di web, video yang durasinya panjang, dan tv/radio *online* yang siaran 24 jam.

2.4 H.264

H.264 (MPEG-4 *Part 10*) atau lebih dikenal dengan *Advance Video Coding* (AVC) merupakan sebuah *codec video digital* yang memiliki keunggulan dalam rasio kompresi (tingkat kompresi yang tinggi) dengan memanfaatkan metoda blok transformasi adaptif yang efektif [7]. H.264 dikembangkan oleh ITU-T *Video Coding Expert Group* (VCEG) bersama-sama dengan ISO/IEC *Moving Picture Expert Group* (MPEG), yang dinamakan *Joint Video Team* (JVC) [7]. Tujuan pengembangan H.264/AVC adalah untuk membuat suatu standar video digital yang dapat menghasilkan kualitas video yang baik pada *bitrate* yang lebih kecil dibandingkan dengan standar video *digital* sebelumnya (MPEG-2, H.263, maupun MPEG-4 *Part 2*) tanpa harus melakukan perubahan yang kompleks, dan dapat diimplementasikan dengan biaya yang murah. Tujuan lain dari pengembangan H.264 adalah dapat digunakan dalam berbagai macam aplikasi seperti video *broadcast*, DVB *storage*, RTP/IP *packet networks*, dan ITU-T *multimedia telephony systems* [7].

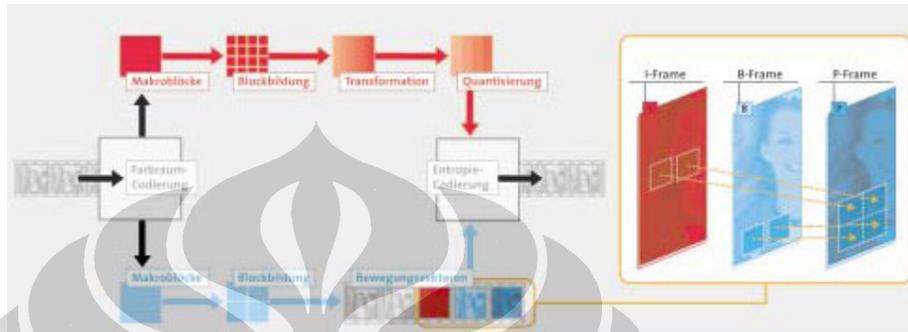
2.4.1 Struktur video

Sebuah film hanyalah sebuah deretan gambar-gambar yang disebut *frame*. Standar TV PAL dapat mengirimkan 25 *frame* per detik. Encoder H.264 akan menggabungkan deretan *frame* ini dalam sebuah deretan terpisah terlebih dahulu yang disebut *Group of Pictures* (GOP) [7]. Sebuah GOP biasanya terdiri atas 12 sampai 15 *frame*. Encoder H.264 membagi-bagi setiap *frame* dalam *macroblock* menjadi 16x16 *pixel*. Dari *macroblock* ini, *encoder* akan menentukan nilai pencahayaan (*luminance*) dan nilai warna (*chromaticity*), serta mendefinisikannya ke dalam blok kecil yang berbeda. H.264 menggunakan 5 tipe ukuran, yaitu potongan tipe *Intra-frame* (*I-frame*), *Predicted-frame* (*P-frame*) dan *Bidirectional-frame* (*B-frame*), yang biasa disebut I, P, dan B. Dua tipe lainnya yaitu, *Switching P* (SP) dan *Switching I* (SI), jarang diproses dan berfungsi untuk mengolah video dengan *bitrate* variabel secara efisien [7].

2.4.2 Proses *Encoding* pada H.264

Pada proses *encoding* berdasarkan lingkup warna, *encoder* memproses *frame* melalui dua cara, seperti dapat dilihat pada Gambar 2.3 [7]. Pada proses *intracode*

(merah), hanya posisi tengah dari *brightness* dan informasi warna saja yang dikompresi. Sedangkan pada proses *intercode* (biru), *encoder*-nya menganalisa bagian-bagian mana saja dari sebuah frame yang mengalami perubahan dibandingkan dengan frame lainnya. Informasi ini digambarkan dalam bentuk sebuah *dynamic vector* [7].



Gambar 2.3 Proses Encoding H.264

Pada metode *intracoding*, gambar pertama (*I-frame*) dalam GOP selalu di-*intracode* [7]. Artinya, *encoder* hanya menggunakan informasi-informasi yang ada dalam gambar untuk proses *encoding*. Ini bisa dilakukan melalui *Separated Integer-Transformation*. Disini, informasi gambar diinversikan menjadi frekuensi dan dioptimalkan menjadi blok berukuran 4×4 *pixel* sehingga kualitas gambar bisa bertambah. Saat proses *quantizing*, frekuensi-frekuensi ini dibulatkan pada seluruh angka sehingga dalam jangkauan frekuensi tinggi akan banyak nilai yang terhapus. Nilai yang telah ditentukan ini akhirnya dikompresi tanpa kehilangan data, melalui metode *Entropy Encoding*.

Pada metode *interceding*, H.264 dapat melakukan hal yang lebih banyak dengan *dynamic vector*. Semakin banyak informasi gambar yang direkam oleh *encoder* mengenai *dynamic vector*, semakin tinggi tingkat kompresi data videonya [7]. *Encoder H.264* tidak hanya dapat mengkontruksi *dynamic vector* dari frame sebelumnya atau setelahnya, melainkan juga dari sejumlah frame yang dipilih, baik letaknya di dalam maupun di belakangnya. Selain itu, H.264 juga dapat menentukan *dynamic vector* dari *B-frame*. Hasilnya, kompresi lebih menghemat ruang, karena semua tipe frame bisa dijadikan sebagai gambar referensi untuk *dynamic vector*.

Proses terakhir adalah *entropy coding*, yang akan mengkompresi data hasil *intracoding*, dan *intercoding* sebelumnya tanpa ada data yang hilang [7]. Prosesnya sama seperti kompresi ZIP sebuah file. Untuk itu, H.264 melakukan dua metode, yaitu *Context Adaptive Variable Length Coding (CAVLC)* sehingga *encoder* menghasilkan lebih banyak *code table* dan kemudian dipilih berdasarkan *data analyze*. Hasilnya, kompresi menjadi optimal [7]. Metode yang lain adalah *Context Adaptive Binary Arithmetic Coding (CABAC)* yang lebih kuat [7]. Disini, tidak ada Tabel yang dihasilkan seperti pada CAVLC. CABAC sudah merupakan Tabel itu sendiri sehingga jadi lebih baik. Dibandingkan dengan CAVLC, CABAC bekerja lebih efisien sekitar 10 sampai 15 persen [7].

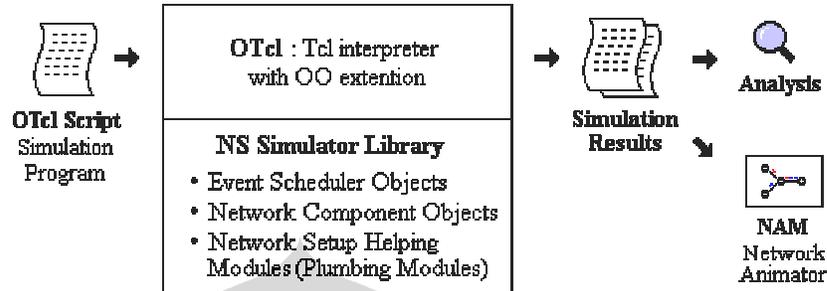
2.5 Network Simulator 2 (NS-2)

Network Simulator-2 (NS-2) pertama kali dikembangkan pada tahun 1995 di University of California Berkeley yang didukung oleh *Defense Advanced Research Projects Agency (DARPA)*. NS-2 merupakan suatu sistem yang bekerja pada *operating system* Unix/Linux. NS-2 juga dapat dijalankan dalam sistem Windows namun harus menggunakan *Cygwin* sebagai *environment* Linux-nya [8].

Pada dasarnya NS-2 adalah *interpreter* OTcl dengan objek library *network simulation*. Simulator ini mendukung hierarki *class* dalam C++ (*compiled hierarchy*) dan hierarki *class* yang serupa pada *interpreter* OTcl (*interpreted hierarchy*) [9]. Dua hierarki ini saling terkait satu sama lain. *Root* dari hierarki ini adalah *TclObject*. *User* membuat obyek Simulator (dari *class Simulator*) baru melalui *interpreter*, dan dicerminkan dengan obyek sebanding pada *compiled hierarchy* [9]. Setelah obyek Simulator dibuat, maka metode-metode untuk membuat topologi, *node*, dan komponen jaringan lainnya dipanggil.

NS-2 juga merupakan simulator yang dipicu oleh *event (event-driven simulator)* [9]. *Scheduler* berjalan dengan cara mengeksekusi *event* berikutnya yang paling dahulu sampai selesai, kemudian kembali menjalankan *event* berikutnya. Walaupun jaringan berkomunikasi dengan melewati paket, namun itu tidak menghabiskan banyak waktu. Jika komponen yang dibuat memerlukan *delay*, maka digunakan *event scheduler* untuk menerbitkan *event* untuk paket

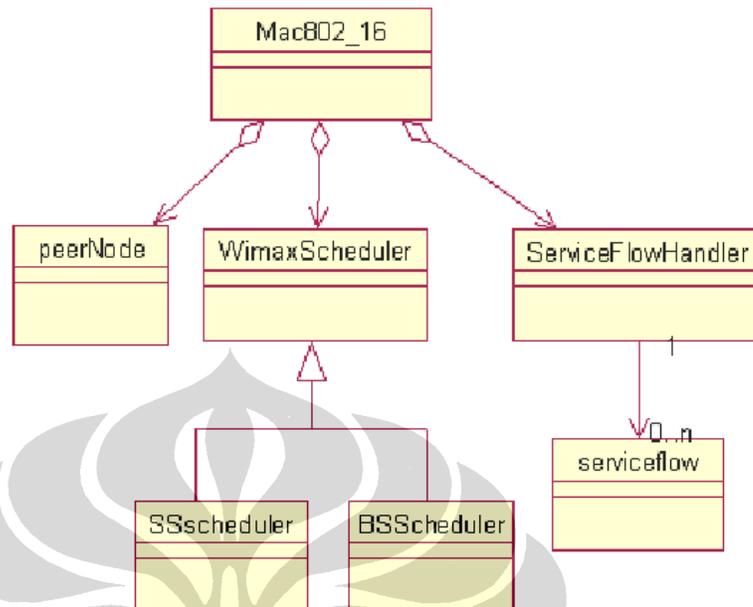
tersebut dan menunggu *event* berhenti sebelum *event* berikutnya dijalankan [9]. Flowchart sistem kerja NS secara global dapat dilihat pada Gambar 2.4.



Gambar 2.4 Flowchart Sistem Kerja NS-2 [9]

2.5.1 Modul QoS-included WiMAX prerelease-09-04-2008

Modul WiMAX yang di kembangkan menggunakan C++ oleh Aymen Belghith ini beroperasi di atas NS 2.29. Modul ini berdasar modul WIMAX *National Institute of Standards and Technology* (NIST). Modul WiMAX NIST mengimplementasikan *layer* OFDM PHY dan TDD MAC [1]. Layer PHY memiliki beberapa parameter yang dapat dikonfigurasi, seperti kekuatan transmisi, *cyclic prefix*, *bandwidth frekuensi*, dan *Modulation Coding Schemes* (MCS)[10]. Modul ini menghitung beberapa nilai seperti frekuensi *sampling*, durasi waktu simbol OFDM dan waktu transmisi untuk sebuah paket berdasar ukuran dan MCS yang digunakan [1].



Gambar 2.5 Diagram kelas MAC IEEE 802.16 MAC [1]

Pada Layer MAC terdapat beberapa *MAC Management Message*. Seperti *Downlink Channel Descriptor (DCD)*, *Uplink Channel Descriptor (UCD)*, *DownlinkMAP (DL-MAP)*, *Up-link MAP (UL-MAP)*, *ranging request*, *ranging response*, *registration request*, dan *registration response* [1]. Sebuah koneksi data *downlink* dan *uplink* dapat ditambahkan untuk tiap *subscriber*. BS menggunakan penjadwal *Round Robin (RR)* untuk mengalokasikan sumber daya radio untuk koneksi *uplink*. WiMAX NIST juga mendukung mobilitas, serta operasi *scanning* dan *handover* [1].

Hal yang ditambahkan oleh Aymen pada modul ini adalah parameter QoS untuk *service flow*, *adaptasi link* dan beberapa algoritma penjadwalan untuk tiga kelas QoS : UGS, rtPS dan BE, selain itu juga ditambahkan *unicast* dan *contention request opportunities mechanism*, seperti yang terdapat dalam standar IEEE 801.16 [1]. Diagram kelas MAC IEEE 802.16 MAC ditunjukkan pada Gambar 2.5.

Modul ini difokuskan pada pengembangan layer MAC 802.16 karena lingkup *scheduler* dan QoS ada pada layer MAC. Kelas MAC 802.16 mewakili MAC *layer*. Ia merupakan kelas utama dan memiliki relasi dengan kelas lain, yaitu :

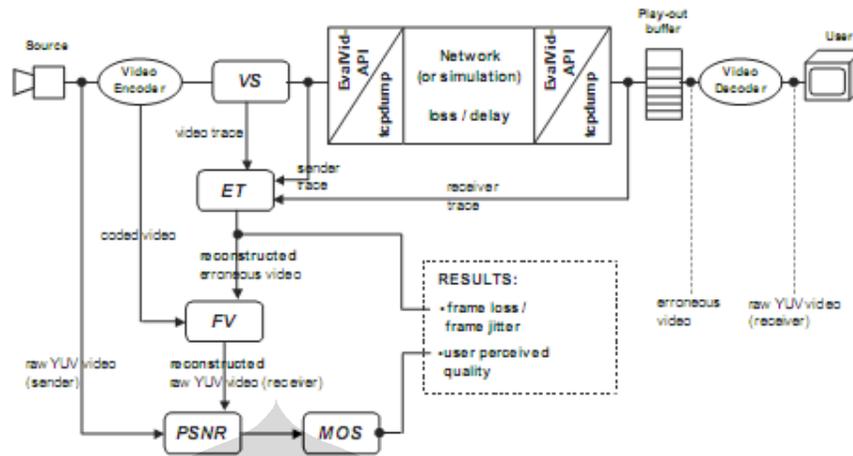
ServiceFlowHandler, *peerNode*, dan *WimaxScheduler* [1]. *ServiceFlowHandler* bertanggung jawab menangani koneksi *downlink* dan *uplink*. Setiap koneksi memiliki sebuah asosiasi dengan *service flow* yang terdapat parameter QoS. Parameter QoS atas sebuah *service flow* ditentukan agar berdasarkan kebutuhan koneksi. Pada *peerNode* terdapat informasi mengenai SS atau BS. *WimaxScheduler* bertanggung jawab untuk pengukuran jarak dan pendaftaran dan melakukan penjadwalan algoritma dimana terdapat dua penjadwal [1]: satu untuk BS (*BSSscheduler*) dan yang lain untuk SS (*SSscheduler*).

2.6 Evaluation Video (EvalVid)

Evaluation Video (EvalVid) merupakan *framework* dan *tool-set* untuk mengevaluasi kualitas video yang ditransmisikan melalui jaringan komunikasi nyata atau simulasi [10]. Evalvid dikembangkan oleh *Technical University of Berlin, Telecommunication Network Group* (TKN) [10]. Evalvid ditargetkan untuk periset yang ingin mengevaluasi perancangan jaringan atau menyusunnya berdasar kualitas video yang dirasakan pengguna.

Disamping mengukur parameter QoS pada jaringan, seperti *loss rate*, *delay*, dan *jitter*. EvalVid juga mendukung evaluasi kualitas *video subyektif* dari video yang diterima berdasarkan perhitungan PSNR *frame-by-frame* [10]. EvalVid menggunakan struktur modular, sehingga pengguna dapat dengan mudah mengganti *codec* dan memungkinkan dilakukannya pertukaran jaringan [10].

Pada struktur *framework* EvalVid diilustrasikan interaksi antara arus data dan tool yang diimplementasikan. *Framework* ini berisi transmisi lengkap dari video digital mulai dari *source video*, *reordering* pada *source*, *encoding*, paketisasi, transmisi jaringan, reduksi *jitter* oleh *buffer play-out*, *decoding*, hingga tampilan video yang diterima oleh *end-user*. Data yang diproses pada arus transmisi akan ditandai dan disimpan pada file-file yang beranekaragam, kemudian file-file ini digunakan untuk memperoleh hasil yang diinginkan, seperti *loss rate*, *jitter*, dan kualitas video. Struktur *framework* EvalVid dapat dilihat pada Gambar 2.6.



Gambar 2.6 Struktur framework evaluasi pada EvalVid

Komponen- komponen utama pada *framework* evaluasi EvalVid dijelaskan secara singkat dibawah ini [11].

- *Source*
Sumber video adalah *raw video* yang berformat YUV QCIF (176 x 144) atau YUV CIF (352 x 288).
- *Video Encoder* dan *Video Decoder*
Saat ini EvalVid mendukung dua macam *codec* MPEG4 yaitu *National Chiao Twig University (NCTU) codec*, *ffmpeg*, *Xvid* dan *H.264*.
- *Video Sender (VS)*
Komponen VS membaca file video yang dikompresi dari output *video encoder*, memfragmentasi masing-masing frame video yang besar menjadi segmen-segmen yang kecil, dan kemudian mengirimkan segmen-segmen ini via paket UDP pada *real network* atau simulasi. Untuk tiap paket UDP yang ditransmisikan, framework merekam *timestamp*, *packet ID*, dan *packet payload size* pada file trace pengirim dengan bantuan *third-party tool*. Untuk *real-network* dapat digunakan *tcp-dump* atau *win_dump*, sedangkan untuk simulasi jaringan dapat digunakan NS-2, Qualnet, atau OPNet. Komponen VS juga *generate* file trace video yang berisi informasi tentang tiap frame pada file video *real*. File *trace* video dan file *trace* pengirim digunakan nanti untuk evaluasi kualitas video berikut.

- *Evaluate Trace (ET)*

Evaluasi berlangsung pada sisi pengirim ketika transmisi video berakhir. Berdasarkan file video asli yang di-*encode*, file *trace* video, file *trace* pengirim dan file *trace* penerima, komponen ET membuat laporan *packet/frame loss* dan *frame/packet jitter* serta mengenerate rekontruksi file video yang berkaitan dengan kemungkinan video *corrupt* pada sisi penerima yang akan direproduksi pada *end-user*.

- *Fix Video (FV)*

Pengujian kualitas video digital dilakukan frame demi frame. Oleh karena itu, jumlah frame video pada sisi penerima termasuk frame *erroneous* harus sama dengan video asli pada sisi pengirim. Jika *codec* tidak dapat menangani frame yang hilang, komponen FV digunakan untuk memecahkan masalah dengan memasukkan frame *decode* di tiap frame yang hilang sebagai metode penyembunyian error.

- *Peak Signal to Noise Ratio (PSNR)*

PSNR merupakan salah satu metrik obyektif yang banyak digunakan untuk menguji QoS pada level aplikasi dari transmisi video.

- *Mean Opinion Score (MOS)*

MOS adalah metrik subyektif untuk mengukur kualitas video digital pada level aplikasi

Pada Gambar 2.6 yang bernama VS, RT, FV, PSNR dan MOS merupakan program sebearnya pada *framework*. Adapun data yang dibutuhkan untuk implementasi pengukuran dengan menggunakan EvalVid antara lain [12]:

1. Dari sisi pengirim

- *Raw uncompressed video*
- *Encoded video*
- *Time-stamp* dan tipe dari tiap paket yang dikirimkan

2. Dari sisi penerima

- *Time-stamp* dan tipe dari tiap paket yang diterima
- *Reassembled encoded video* (kemungkinan *erroneous*)
- *Raw uncompressed video* untuk *display*

Untuk menggunakan tool dalam EvalVid yang dibutuhkan hanyalah file-file *trace*, file video asli dan *decoder* [12]. Kemudian konteks jaringan dari EvalVid hanyalah “*blackbox*” yang meng-*generate delay*, *loss* dan *reordering* paket yang mungkin. Jaringan tersebut dapat berupa *real link* seperti *Ethernet* atau *Wireless Local Area Network (WLAN)* maupun simulasi atau emulasi dari jaringan [12]. Karena hanya interaksi dari EvalVid dan jaringan yang direpresentasikan oleh dua file *trace* (pengirim dan penerima), isi *network box* dapat dengan mudah digantikan, hal ini membuat EvalVid sangat fleksibel [12]. Demikian halnya dengan *codec video* yang dapat juga dengan mudah digantikan.

2.7 Parameter Kinerja

Untuk mengukur kualitas dari performa layanan *video streaming*, dibutuhkan parameter-parameter QoS dan parameter-parameter kualitas video yang mendukung. Parameter tersebut antara lain *frame loss*, *delay*, dan *jitter* untuk parameter QoS, serta PSNR dan MOS untuk parameter kualitas video [12].

2.7.1 Frame Loss

Sebuah frame video (sebenarnya menjadi *single coded image*) relatif berukuran besar. Tidak hanya pada kasus video VBR, tetapi juga CBR, karena masa tetap mengaplikasikan pada waktu pendek rata-rata. *I-frame* biasanya dianggap lebih besar dari target (*short time average*) CBR bahkan pada video CBR [12].

Sangatlah mungkin beberapa atau kemungkinan seluruh frame lebih besar dibandingkan *maximum transfer unit (MTU)* pada jaringan [12]. Ini adalah ukuran paket maksimum yang didukung oleh jaringan (seperti *Ethernet* = 1500 dan 802.11b WLAN = 2312 bytes) [12]. Frame-frame ini harus disegmentasi pada paket yang lebih kecil yang cocok pada jaringan MTU. Kemungkinan segmentasi frame ini menimbulkan masalah perhitungan dari *frame loss*. Pada dasarnya *frame loss rate* dapat diperoleh dari *packet lose rate*. Tetapi proses ini tergantung kapabilitas dari kegunaan *decoder* video aktual, karena beberapa *decoder* dapat memproses frame walaupun beberapa bagian hilang. Lebih lanjut, frame dapat di-*decode* tergantung pada dimana paket hilang. Jika paket pertama hilang, frame

hampir tidak bisa di-*decode*. Kemudian, kapabilitas dari beberapa *decode* harus dipertimbangkan dengan tujuan untuk menghitung *frame loss rate*. Persamaan 2.1 menghitung tipe frame masing-masing [12].

$$FLT = 100 \frac{nT_{recv}}{nT_{sent}} \quad (2.1)$$

Dimana T : Type data frame (salah satu dari *header*, I, P, B, SP, SI)

nT_{sent} : Jumlah type paket T yang dikirim

nT_{recv} : Jumlah type paket T yang diterima

2.7.2 Delay dan Jitter

Ukuran *delay* penerimaan paket yang melambangkan *smoothness* dari *audio/video playback* [12]. Pada sistem transmisi video tidak hanya *loss* yang penting untuk kualitas video yang dirasakan, tetapi juga *delay* dari frame dan *delay variation (frame jitter)*. Video digital selalu terdiri dari frame yang harus ditampilkan pada *constant rate*. Menampilkan frame pada sebelum atau setelah waktu yang didefinisikan menghasilkan “ *jerkiness*”. Isu ini dialamatkan oleh yang dinamakan *play-out buffer*. *Buffer* ini memiliki tujuan menyerap *jitter* yang ditimbulkan oleh *delay* penyampaian *network*. Hal ini jelas *play-out buffer* yang cukup besar yang dapat mengkompresi sejumlah *jitter*. Pada kasus yang ekstrim, *buffer* dapat sebesar seluruh video, dan melakukan awalan tidak setelah frame terakhir diterima. Hal ini akan mengeliminasi beberapa kemungkinan *jitter* pada waktu *delay* tambahan dari seluruh waktu transmisi. Hal ekstrim lainnya yaitu kapabilitas *buffer* dalam memegang 1 frame dengan tepat. Pada kasus ini tidak ada *jitter* sama sekali yang dapat dieliminasi tetapi tidak ada *delay* tambahan yang ditimbulkan.

Terdapat metode yang dikembangkan untuk mengoptimisasi *play-out buffer* berkenaan dengan *particular trade-off* [12]. Metode ini tidak di dalam cakupan *framework* yang dijelaskan. Ukuran *play-out buffer* hanya sebuah parameter pada proses evaluasi. Hal ini terbatas pada *framework play-out buffer static* [12].

Bagaimanapun, karena strategi integrasi *play-out buffer* pada proses evaluasi, *loss* tambahan dikarenakan *play-out buffer* dapat dipertimbangkan.

Definisi formal *jitter* dapat dilihat pada Persamaan 2.2 [12]. Ini adalah variasi dari waktu *inter-packet* atau *inter-frame*. "*Frame time*" ditentukan oleh waktu dimana segmen terakhir dari frame yang telah tersegmentasi diterima.

inter-packet time (2.2)

$$it_{P_0} = 0$$

$$it_{P_n} = t_{P_n} - t_{P_{n-1}}$$

dimana t_{pn} : *time-stamp* dari paket n

inter-frame time

$$it_{F_0} = 0$$

$$it_{F_m} = t_{F_m} - t_{F_{m-1}}$$

dimana t_{Fm} : *time-stamp* dari segmen terakhir dari frame m

packet jitter

(2.3)

$$j_P = \frac{1}{N} \sum_{i=1}^N (it_i - it_N)^2$$

dimana N : paket N

it_N = rata-rata dari waktu *inter-packet*

frame jitter

(2.4)

$$j_F = \frac{1}{M} \sum_{i=1}^M (it_i - it_M)^2$$

dimana N : frame N

it_N = rata-rata dari waktu *inter-frame*

2.7.3 Evaluasi Kualitas Video

Pengukuran kualitas video digital harus berdasarkan pada kualitas yang dirasakan pada video aktual yang sedang diterima oleh user pada sistem digital

video, karena kesan *user* adalah apa yang dihitung di akhir. Terdapat dua pendekatan dasar untuk mengukur kualitas video digital, yaitu pengukuran kualitas subyektif dan pengukuran kualitas obyektif. Pengukuran kualitas subyektif selalu merenggut faktor krusial, kesan user melihat video ketika sedang berhemat, menghabiskan banyak waktu, kebutuhan sumberdaya manusia yang tinggi dan perlengkapan khusus yang dibutuhkan. Beberapa metode obyektif dijelaskan secara detail pada ITU, ANSI, dan MPEG [12]. *Human quality impression* biasanya diberikan pada skala dari 5 (terbaik) ke 1 (terburuk) sebagaimana terlihat pada Tabel 2.2 [12]. Skala ini dinamakan *Mean Opinion Score* (MOS).

Tabel 2.2 Kualitas dan Tingkat Kerusakan ITU-R [12]

Skala	Kualitas	Kerusakan
5	Excellent	Imperceptible
4	Good	Perceptible, but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

Beberapa *task* di industri dan penelitian membutuhkan metode otomatis untuk mengevaluasi kualitas video. Tes subyektif yang kompleks dan mahal seringkali tidak dapat diterima. Oleh karena itu, *objective metric* telah dikembangkan untuk mengemulasi *quality impression* dari *human visual system* (HVS) [12]. Terdapat diskusi mendalam dari berbagai macam *objective metric* dan performa mereka dibandingkan dengan tes subyektif.

2.7.3.1 Peak Signal to Noise Ratio (PSNR)

Metode pengukuran kualitas video yang tersebar luas adalah perhitungan dari *peak signal to noise ratio* (PSNR) gambar demi gambar [12]. PSNR merupakan turunan dari *signal to noise ratio* (SNR) yang membandingkan sinyal energi dengan error energi. PSNR merupakan dasar dari *quality metric* yang digunakan pada framework untuk menguji hasil dari kualitas video [12].

PSNR membandingkan kemungkinan maksimum sinyal energi dengan error energi, dimana telah memperlihatkan hasil pada korelasi yang lebih tinggi dengan

persepsi kualitas subyektif dengan SNR yang konvensional. Persamaan 2.5 adalah definisi dari PSNR antara komponen *luminance* Y dari gambar sumber S dan gambar tujuan D [12].

$$PSNR_{(n)dB} = 20 \log_{10} \left(\frac{V_{peak}}{\sqrt{\frac{1}{N_{col}N_{row}} \sum_{i=0}^{N_{col}} \sum_{j=0}^{N_{row}} [Y_S(n,i,j) - Y_D(n,i,j)]^2}} \right) \quad (2.5)$$

dimana $V_{peak} = 2^k - 1$

k : jumlah bit per *pixel* (komponen *luminance*)

Bagian di penyebut merupakan *mean square error* (MSE). Formula untuk PSNR dapat diringkas sebagai $PSNR = 20 \log \frac{V_{peak}}{MSE}$ [12]. Karena PSNR dihitung berdasarkan *frame by frame* dapat membuat ketidaknyamanan, ketika diaplikasikan pada video yang terdiri dari beberapa ratus atau ribu frame. Lebih lanjut, orang-orang lebih tertarik pada distorsi yang diperkenalkan oleh jaringan sendiri. Jadi mereka ingin membandingkan video yang diterima (kemungkinan distorsi) dengan video yang dikirimkan yang tidak terdistorsi. Hal ini dapat dilakukan dengan membandingkan PSNR dengan *envoked video* dengan frame video yang diterima *frame by frame* atau membandingkan rata-rata mereka dan standar deviasi [12].

2.7.3.2 Mean Opinion Score (MOS)

Oleh karena *time series* PSNR sangat tidak ringkas, *metric* tambahan disediakan. PSNR dari tiap frame tunggal akan dipetakan pada skala MOS seperti terlihat pada Tabel 2.3 [12]. MOS merupakan *human impression* dari kualitas video, dimana diberikan pada skala 5 ke 1. Skala 5 menunjuk pada kualitas terbaik, sedangkan skala 1 untuk kualitas terburuk. Metode ini memiliki keuntungan untuk memperlihatkan dengan jelas distorsi yang disebabkan oleh *network at a glance*.

Untuk memperoleh MOS, nilai PSNR pada video yang diberikan ditentukan dengan menggunakan program PSNR, yang kemudian memetakannya pada skala MOS. Program MOS digunakan untuk melakukan pemetaan. PSNR menghitung kemungkinan maksimum *signal energy to noise energy*, dimana secara matematis sama dengan *root mean squared error* [12].

Tabel 2.3 Konversi PSNR ke MOS [12]

PSNR [dB]	MOS
> 37	5 (Excellent)
31 – 37	4 (Good)
25 – 31	3 (Fair)
20 – 25	2 (Poor)
< 20	1 (Bad)