

# PERANCANGAN MEDIA KOMPRESI VIDEO BERBASIS *FAST MODE DECISION ALGORITHM*

Noor Arifin , Sri Heranurweni, Ari Endang Jayati

Program Studi Teknik Elektro, Universitas Semarang.  
Jl. Soekarno Hatta, Semarang 50196, Indonesia

## Abstrak

Kompresi video digital sangat dibutuhkan pada media video digital untuk menghemat ruang penyimpanan dan bandwidth yang besar tanpa mengurangi kualitas video yang terlalu signifikan. Sehingga dibutuhkan sebuah media kompresi yang mampu mengurangi ukuran file video agar lebih efisien. Untuk merancang media kompresi video pada penelitian ini digunakan bahasa pemrograman C# dengan format keluaran file video berupa .MP4 dan MKV. Hasil kompresi dianalisa nilai MSE dan PSNR pada frame pertama. Hasil pengujian menunjukkan nilai PSNR berada pada kisaran 30dB sampai 50dB sesuai standar yang diperbolehkan ITU-T 2014.

**Kata kunci :** *kompresi, video, PSNR.*

## PENDAHULUAN

*High Efficiency Video Coding* (HEVC) sudah diperkenalkan sejak tahun 2014 yang lalu dan merupakan sebuah pengembangan teknik video kompresi terbaru. Media kompresi merupakan media pembelajaran yang sangat diperlukan dalam dunia telekomunikasi dan informasi. Saat ini, media kompresi video telah mengalami evolusi yang sangat pesat.

Saat ini telah banyak pengembangan algoritma untuk merancang suatu aplikasi kompresi video dengan menggunakan sebuah *codec H.265*. Salah satu algoritma yang digunakan adalah *Fast Mode Decision Transcoder Algorithm* Suatu media kompresi yang mampu memperkecil ukuran data video dengan kualitas yang tidak terlalu berbeda dengan data asli video tersebut akan bermanfaat untuk melakukan proses kompresi video dengan lebih efisien.

Video atau gambar bergerak adalah data digital yang terdiri dari beberapa gambar. Istilah video biasanya mengacu pada beberapa format penyimpanan gambar bergerak. Terbagi menjadi dua, yaitu video analog, contohnya VHS dan Betamax, dan

video digital, contohnya DVD, *Quicktime*, dan *MPEG-4*. Video dapat direkam dan ditransmisikan dalam berbagai media fisik, pada pita magnetik ketika direkam sebagai Pal atau NTSC *signal* elektrik dengan video kamera, atau *MPEG-4* ketika direkam menggunakan kamera digital[3].



Gambar 1. Contoh Frame Video Digital [3]

## TINJAUAN PUSTAKA

H.265 merupakan singkatan dari *High Efficiency Video Coding*(HEVC), menawarkan kualitas video yang setara dengan codec yang ada saat ini yaitu H.264 dengan bandwidth hanya setengah yang dibutuhkan codec lama tersebut. Hal itu tentunya akan membawa keuntungan tersendiri dan membuka berbagai kemungkinan untuk penyedia konten

berbasis video. Codec baru tersebut memungkinkan ukuran file yang jauh lebih kecil untuk film dengan kualitas yang sama. Atau bila dilihat dari sisi lain, dengan ukuran file yang sama, video bisa memiliki resolusi yang lebih tinggi.

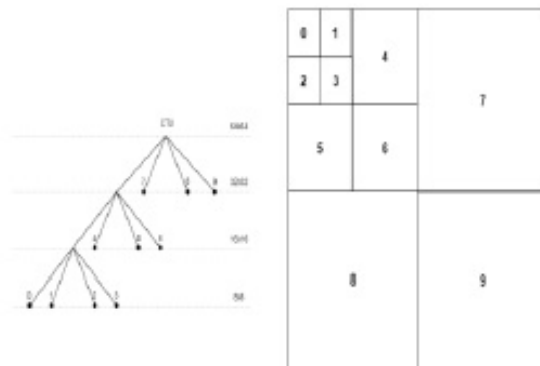
Tiga blok konsep yang berdiri sendiri telah didefinisikan didalam HEVC: CU (*Coding Unit*), PU (*Prediction Unit*) and TU (*Transform Unit*). CU adalah unit dasar dalam pengkodean, CU yang terbesar disebut CTU (*Coding Tree Unit*). CU bisa diprediksi oleh satu PU atau banyak tetapi tidak dapat disamakan dengan PU, yang mana setiap komponen memiliki parameter prediksi tersendiri. Didalam *intra coding*, hanya CU terkecil yang diijinkan untuk diprediksi oleh empat kotak PU. TU adalah unit dasar untuk transformasi [4].

### Coding Tree Unit

Sebuah pengkodean yang menjadi pokok standar menuju HEVC berdasarkan sebuah unit yang dinamakan *macroblock*. Sebuah *macroblock* tergabung dari  $16 \times 16$  pixels yang menetapkan dasar terbentuknya code struktur dari frame yang lebih besar. Konsep ini kemudian diterjemahkan kedalam *Coding Tree Unit (CTU)* dengan standar HEVC, struktur ini fleksibel pada *macroblock*. sebuah CTU biasanya berukuran  $64 \times 64$ ,  $32 \times 32$ , or  $16 \times 16$  pixels.

### Coding Unit

Setiap CTU diatur didalam sebuah kotak segiempat yang selanjutnya dipartisi ke ukuran lebihkecil bernama *Coding Unit (CU)*. Sebagai contoh pempartisian sebuah CTU menuju CU ditunjukkan sebagai berikut:



Gambar 2. HEVC Coding Tree Unit (Sina, 2013)

Diagram pohon adalah garis yang melintang di kedalaman pertama dan sesuai dengan titik diagram pohon terlihat pada struktur CTU digambar 2.

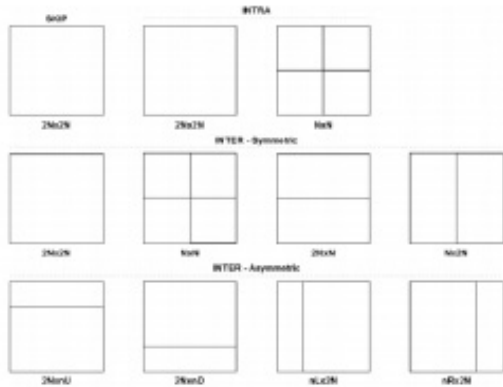
### Prediction Mode

Setiap CU dapat diprediksi menggunakan tiga mode prediksi 1) *Intra-predicted CU*; 2) *Inter-predicted CU*; 3) dan *Skipped CU*. Intra-prediksi menggunakan informasi piksel yang tersedia pada gambar yang bersangkutan sebagai referensi prediksi, dan sebuah arah prediksi ditentukan. *Inter-prediction* menggunakan informasi piksel pada frame sebelum dan sesudahnya sebagai referensi prediksi. *Skipped CU* hampir sama dengan hasil *inter-prediction CU*, hanya saja tidak terdapat informasi *motion*, oleh sebab *skipped CU* menggunakan kembali informasi *motion* yang sudah ada dari frame dan sesudahnya.

Dilihat dari arah prediksinya, AVC mempunyai delapan arah prediksi yang memungkinkan dalam intra blok diAVC, sedangkan HEVC mendukung 34 intra prediksi mode dengan 33 arah yang berbeda, dan untuk mengetahui itu, intra prediksi blok seharusnya berukuran jarak dari  $4 \times 4$  sampai  $32 \times 32$ , ada 132 kombinasi ukuran blok dan arah prediksi yang ditentukan untuk HEVC bit-stream.

### Unit Prediksi

Sebuah cabang CU didalam CTU dapat dibagi lebih jauh lagi menjadi bagian prediksi yang dinamakan *Prediction Units* (PU). sebuah CU dapat dibagi menjadi satu, dua, atau empat PU. PU yang memungkinkan tergantung pada prediksi mode. Untuk intra-prediksi dapat diselesaikan dengan dua mode yang memungkinkan, dimana interprediksi dapat diselesaikan dengan satu dari delapan mode yang memungkinkan. Gambar 3 memperlihatkan mode PU yang memungkinkan diaplikasikan diHEVC dimana N menentukan angka piksel didalam blok.



Gambar 3. Prediksi Unit Mode Partisi (Sina, 2013)

### Transform Unit

Sebuah *Transform Unit* (TU) didefinisikan untuk setiap CU dengan cara yang sama untuk PU dan diatur dalam bentuk *quad-tree*. Setiap TU bertanggung jawab untuk transformasi residu dari PU. Untuk setiap TU, koefisien residu dihitung dengan menerapkan transformasi satu bilangan bulat.

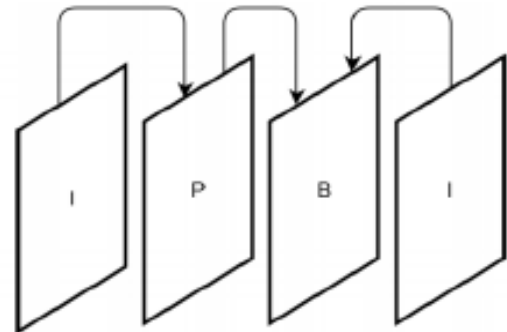
### Motion Vector

Sebuah *Motion Vector* (MV) digunakan untuk inter prediksi dan menentukan dengan blok referensi ke blok sekarang yang berada pada sebelum atau sesudah gambar. MV ditentukan per PU dan setiap PU referensi satu atau dua blok.

### Gambar dan Irisan

Sebuah gambar adalah tertangkapnya sebuah frame dalam waktu t dan dapat dibagi menjadi satu atau lebih irisan. Irisan dikodekan secara terpisah dari bagian lain dari gambar. Irisan dan frame memiliki tiga tipe: I, P, dan B. Dalam irisan-I hanya intra prediksi yang diperbolehkan.

Dalam irisan-P intra prediksi dan inter prediksi dapat digunakan. Tapi hanya satu MV per PU yang diperbolehkan untuk inter-prediksi. Irisan-B memungkinkan inter dan intra prediksi dengan satu atau dua MV per PU.



Gambar 4. Tipe Gambar HEVC dan Hubungan Setiap Tipe (Sina, 2013)

### Pengukuran Kualitas Video

Sangat penting untuk mengkompresi sinyal video sebanyak mungkin dan menjaga kualitas video dekat dengan aslinya. Umumnya, ada dua kategori metode untuk mengukur kualitas video:

1. Kualitas subjektif didasarkan pada prosedur pengujian yang dirancang oleh ITU, sebagaimana tercantum dalam BT.500-11 (2002), untuk mengukur kualitas menggunakan agen manusia.
2. Kualitas obyektif diukur menggunakan model matematika untuk mendekati kualitas subjektif.

Penilaian kualitas subjektif dari video digital membutuhkan agen manusia, namun memerlukan biaya yang cukup besar, maka kualitas obyektif adalah alternatif yang sesuai.

Model matematika yang utama digunakan oleh peneliti adalah *Rate-*

*Distortion Optimization (RDO) Model* (Sullivan dan Wiegand, 1998). Berdasarkan model ini, distorsi dioptimalkan berdasarkan perubahan *rate* yang jumlah data dibutuhkan untuk pengkodean data masukan. Dalam pengkodean video, setiap keputusan biasanya mempengaruhi nilai *Rate-Distortion* (RD), dan tantangannya adalah untuk menemukan solusi optimal. Kriteria RD umum digunakan dalam pengkodean video adalah PSNR-bit rate pair. [10]

### Peak Signal to Noise Ratio

Pengukuran secara objektif pada kualitas video yang biasa digunakan adalah dengan *Peak Signal-to-Noise Ratio* (PSNR). Diukur dalam satuan decibel (dB) seperti berikut : [10]

$$PSNR(\text{img1}, \text{img2}) = 10 \log_{10} \frac{(2^{n-1})^2}{MSE(\text{img1}, \text{img2})}$$

(1)

dimana

PSNR =Level kualitas dalam dB(decibel) stream video hasil kompresi

MSE =Tingkat error antara 2 gambar asli dan terkompresi

Seperti ditunjukkan dalam Persamaan diatas, PSNR diukur berdasarkan *Mean Square Error* (MSE) antara dua gambar, yang mana satu gambar adalah gambar asli dan yang lain adalah gambar terkompresi. *n* adalah jumlah bit yang digunakan untuk mewakili setiap pixel, yang biasanya bernilai 8 bit.

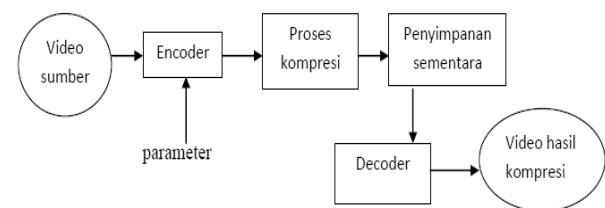
Nilai-nilai PSNR yang tinggi menunjukkan bahwa input dan output gambar yang sama. Nilai PSNR yang baik berkisar dari 30 dB sampai 50 dB. Perhatikan bahwa hanya PSNR untuk saluran luma yang diukur (Y-PSNR).

### Bit Rate

*Bit-rate* (R) dari *bit-stream* dihitung dengan rata-rata jumlah bit didalam *bit-stream* dengan panjang *bitstream* diukur dalam satuan detik. Hasil biasanya diukur

dengan kilobit per detik (kbit / s) atau megabit per detik (Mbit / s). Metode umum untuk mengontrol *bit-rate* oleh *encoder* adalah untuk menyesuaikan *Parameter Quantization* (QP). QP menentukan bagaimana menghitung *quantizer* akan mendekati koefisien. QP semakin tinggi maka akan mengurangi *bit-rate* dan QP sebaliknya lebih rendah akan meningkatkan *bit rate*.

Ringkasan kerja program pada *transcoding* ditunjukkan seperti berikut:



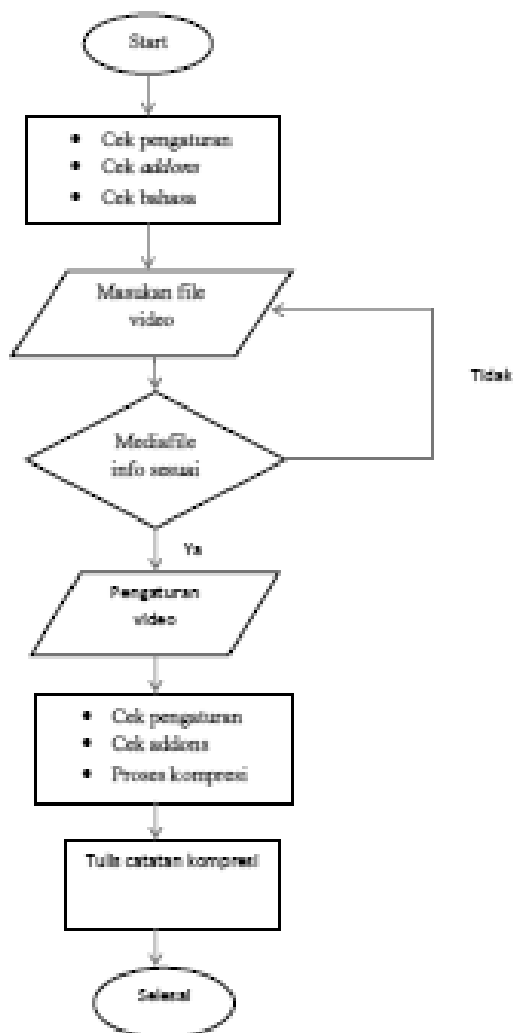
Gambar 5. Blok Diagram Video Transcoding (Ahmad, 2005)

Gambar 5 terlihat bahwa dari video sumber akan diencoding dengan menggunakan parameter-parameter yang ditentukan. Setelah ditentukan parameter yang sesuai, maka akan dilakukan *transcoding* atau proses kompresi yang selanjutnya akan dilakukan penyimpanan sementara. Selanjutnya video akan *didecoding* menjadi file video hasil kompresi.

### FFMPEG

**FFMPEG** adalah program komputer yang dapat perekam, mengkonversikan dan streaming audio dan video digital dalam berbagai format. **FFMPEG** merupakan aplikasi *command line* yang terdiri dari kumpulan pustaka perangkat lunak bebas / *open source*. Termasuk *libavcodec*, *library* untuk *audio codec* / *video codec* yang digunakan oleh beberapa proyek lain, dan *libavformat*, *library* untuk audio / video mux kontainer dan demux kontainer. Nama proyek yang berasal dari grup video standar MPEG, di tambahkan "FF" untuk "*fast forward*"[3].

## metodologi penelitian



Gambar. 6. Diagram Alur Penelitian

Untuk penelitian yang dilakukan, program mendeteksi *addons*, bahasa dan pengaturan sebelumnya jika ada. Namun jika baru pertama kali dibuka, maka program menggunakan pengaturan standar yang diatur oleh penulis. Pengguna memasukkan video digital yang akan dikompresi yang kemudian akan di cek oleh program sesuai dengan mediafile info yang ada pada video digital. Pengguna dapat mengatur hasil kompresi berdasarkan

*ratefactor* pada video dan pengaturan audio yang tersedia. Setelah selesai mengatur sesuai yang diinginkan pengguna maka proses kompresi akan dieksekusi oleh program menggunakan konsol pada *FFMPEG*. Output yang dihasilkan pada program ini berformat MP4 atau MKV sesuai keinginan pengguna.

## analisa dan pembahasan

Percobaan yang dilakukan dengan menggunakan aplikasi yang telah dibuat menghasilkan data sebagai berikut:

TABEL 1  
DATA HASIL VIDEO KOMPRESI

Nama Video	Ukuran Video (MB)	Durasi (S)	Video Bit Rate (kbps)	Piksel
Terkompresi <i>Basketball_1920x1080_30.mp4</i>	6.10	10	5119	1920X1080
Terkompresi <i>RaceHorses_832x480_30.mp4</i>	1.89	10	1592	832X480
Terkompresi <i>SlideEditing_1280x720_30.mp4</i>	1.02	10	856	1280X720
Terkompresi <i>Speed_1024x768_30.mp4</i>	5.80	17	3042	1024X768
Terkompresi <i>Square_416x240_60.mp4</i>	0.464	10	380	416X240
Terkompresi <i>VideoConf_1280x720_60.mp4</i>	0.926	10	758	1280X720

Dalam Tabel 1 terlihat bahwa ukuran file video berbanding lurus dengan ukuran *bitrate* video dalam proses kompresi video. Bila ukuran file semakin kecil, maka harga *bitrate* juga semakin kecil. Durasi yang dihasilkan pada kompresi video sama seperti video asal.

Kesamaan durasi hasil video dengan video asal menandakan bahwa informasi file yang dihasilkan tidak

berbeda dengan file asal. Percobaan yang dilakukan dalam tugas akhir ini, analisa *MSE* dan *PSNR video stream HEVC* yang dihasilkan dapat diamati dengan menggunakan program *Elecard Quality Estimator*.

Nilai MSE merupakan nilai rata-rata selisih kuadrat antara nilai piksel pada frame referensi dikurangi dengan nilai piksel frame yang diproses. Sedangkan nilai PSNR merupakan nilai logaritmik yang tergantung dari hasil MSE dengan satuan dB(Desibel). Analisa tugas akhir ini hanya diamati nilai MSE dan PSNR pada frame pertama. Video kompresi dengan format .mp4 didapatkan hasil sebagai berikut:

TABEL 2  
DATA HASIL PERHITUNGAN MSE VIDEO  
KOMPRESI

NO	File Video Digital	Nilai MSE			
		Y	U	V	Rata-rata
1	Terkompresi <i>Basketball_1920x1080_50.mp4</i>	10.6835	5.8787	6.9024	7.8215
2	Terkompresi <i>RaceHorses_832x480_30.mp4</i>	54.4696	60.4250	54.6054	56.5000
3	Terkompresi <i>SlideEditing_1280x720_30.mp4</i>	7.6747	22.0416	23.3197	17.6787
4	Terkompresi <i>Speed_1024x768_30.mp4</i>	24.4639	29.5001	42.1699	32.0447
5	Terkompresi <i>Square_416x240_60.mp4</i>	119.1241	114.1265	104.2108	112.4871
6	Terkompresi <i>VideoConf_1280x720_60.mp4</i>	5.3435	3.7895	3.4259	4.1863

Dari Tabel 2 dapat diketahui bahwa nilai MSE terbesar ada pada 7 video file video *square\_416x240\_60.mp4*, dikarenakan tingkat perubahan motion image yang tidak terlalu signifikan, sehingga dapat disimpulkan bahwa MSE dipengaruhi oleh tingkat pengaturan *bitrate*, dan tingkat kinerja *motion compensation block*.

TABEL 3  
DATA HASIL PERHITUNGAN PSNR VIDEO  
KOMPRESI

NO	File Video Digital	Nilai PSNR			
		Y	U	V	Rata-rata
1	Terkompresi <i>Basketball_1920x1080_50.mp4</i>	40.9053	46.5102	46.8607	42.8966
2	Terkompresi <i>RaceHorses_832x480_30.mp4</i>	37.7659	40.7654	40.3256	38.9351
3	Terkompresi <i>SlideEditing_1280x720_30.mp4</i>	39.9057	39.4444	39.6198	39.7145
4	Terkompresi <i>Speed_1024x768_30.mp4</i>	43.9364	40.9603	42.9367	42.9367
5	Terkompresi <i>Square_416x240_60.mp4</i>	40.3881	43.9792	43.5845	41.7602
6	Terkompresi <i>VideoConf_1280x720_60.mp4</i>	44.2174	49.1582	48.7201	45.9645

Dari Tabel 3 dapat disimpulkan bahwa nilai PSNR file video hasil kompresi berada pada kisaran 30dB sampai 50dB, hal ini membuktikan bahwa *codec H.265* memenuhi standar video kompresi sesuai ITU-T tahun 2014 tentang standar video kompresi yaitu berada pada 30dB sampai 50dB.

### Kesimpulan

Hasil video kompresi yang telah diuji mempunyai nilai PSNR yang berada pada range 30dB sampai 50dB. Telah memenuhi standar video kompresi sesuai ITU-T tentang video kompresi tahun 2013. Kualitas kompresi sangat dipengaruhi oleh *bitrate* video. Semakin rendah nilai *bitrate* video, maka semakin rendah pula kualitas video tersebut. Pada perancangan program kompresi video telah melakukan kompresi frame pada video digital mencapai 50% sampai 85% dari frame sumbernya namun tetap menjaga kualitas gambar hasil kompresi. Nilai MSE pada video *square* cenderung tinggi dikarenakan variasi perubahan *motion image* tidak terlalu signifikan, sehingga dapat disimpulkan bahwa MSE (*Mean Squared Error*) sangat

dipengaruhi oleh kinerja dari *motion compensation block*.

#### Daftar Pustaka

2015. "Format Baru H265 Standar Video Codecs". Available :<http://if-unsika-2010214.blogspot.com/2013/06/format-baru-h265-standarvideo-codecs.html>.
2015. "Pengertian Audio dan Media Audio Secara Lengkap". Available : <http://www.pengertianku.net/2014/11/pengertian-audio-anmedia-audio-secaralengkap.html>.
2015. "Pengertian Video". Available : <http://diyarblablابلap.blogspot.com/2012/06/pengertianvideo>.
2015. Available : <http://www.itu.int/rec/T-RECH.265-201410-I/en>.
- B. Bross. "An overview of the next generation High Efficiency Video Coding(HEVC), in Next Generation Mobile Broadcasting", (ed. David Gómez-Barquero), CRC Press. 2013.
- Enhorn, Jack. "Efficient Video Coding Beyond High-Definition Resolution", Stockholm, Sweden.2011.
- HEVC to H.264/AVC Transcoding with Fast Intra Mode Decision", in *Proc. The 19th International Conference on Multimedia Modeling*. 2013.
- [html?m=l](#).
- ITU-T Rec. H.265 dan ISO/IEC 23008-2. "Highefficiency video coding". ITU-T dan ISO/IEC JTC 1. Versi 1. 2014.
- J. Zhang, F. Dai, Y. Zhang, dan C. Yan. "Efficient
- Setiabudi,Wahyu, "Pengembangan Teknik Kompresi Video Digital Dengan Menggunakan *Motion Compensation* Berbasis Algoritma SAD (*sumabsolute differences*) dan 2DDCT (*2-dimension discretecosine transform*)", FT UI. 2008.
- T. Wiegand, W.-J. Han, B.Bross, J.-R. Ohm, dan G. J.Sullivan, Oct. "WorkingDraft 1 of High Efficiency Video Coding, ITU-T/ISO/IEC Joint Collaborative

Team on Video Coding (JCT-VC)", *document JCTVC-C402*. 2010.

Wang, Ying. "Analysis Application for H.264 VideoEncoding", UPPSALA Universitet. 2010.