Data Compression - Lossy and Lossless Compression

Mathias Novianto - 13516021
Ferdiant Joshua M - 13516047

Data compression, atau *bit-rate reduction* adalah metode *encoding* suatu informasi agar jumlah bit yang dipakai lebih sedikit dari representasi aslinya. *Data compression* dilakukan untuk mengefektifkan penggunaan *storage space*. Seluruh bit data yang dapat diperkecil akan dikompres semaksimal mungkin agar diperoleh *file* yang sedekat mungkin dengan *file* aslinya tetapi dengan ukuran sekecil mungkin.

Teknik pengecilan ukuran *file* yang paling umum dilakukan adalah menghapus dan mengganti data elemen yang repetitif. *Data compression* dapat dilakukan secara *lossless* (tidak ada data yang hilang) dan juga *lossy* (data yang tidak begitu signifikan dihapus). Berikut adalah penjelasan lebih lanjut mengenai *lossy compression* dan *lossless compression*.

A. Lossy Compression

Lossy compression adalah sebuah teknik kompresi data dengan menghilangkan sebagian informasi yang hasil akhirnya masih dapat diterima. *Lossy compression* umumnya dipraktikkan pada gambar, suara, dan video. Metode ini memanfaatkan keterbatasan indera manusia (mata, telinga) dalam melihat dan membaca detil suatu gambar, suara, atau video. Indera manusia secara umum tidak dapat membedakan secara eksak *file* sesudah dan sebelum dilakukannya *lossy compression* selama metode tersebut tidak dilakukan secara berlebihan. Perbedaan dan pengurangan kualitas dapat terlihat ketika *file* yang dikompres dilihat lebih detil dengan komputer.

Lossy compression menghilangkan sebagian informasi dari suatu *file*, oleh karena itu ukuran *file* yang diperoleh dapat lebih kecil hingga 80% (misalnya untuk gambar dengan format JPEG). Metode kompresi ini memberikan pilihan kepada pengguna untuk memiliki *file* dengan kualitas yang tinggi tetapi ukurannya besar, atau kualitasnya berkurang (yang umumnya tidak begitu dapat dibedakan indera manusia) tetapi hemat tempat karena ukurannya lebih kecil. Berikut adalah beberapa metode *lossy compression* :

- JPEG

  JPEG (Joint Photography Expert Group) adalah sebuah metode kompresi yang menjadi standard pada tahun 1992. Metode ini secara garis besar terbagi menjadi 5 tahap, yaitu :

  i. Konversi warna dari RGB ke YCbCr

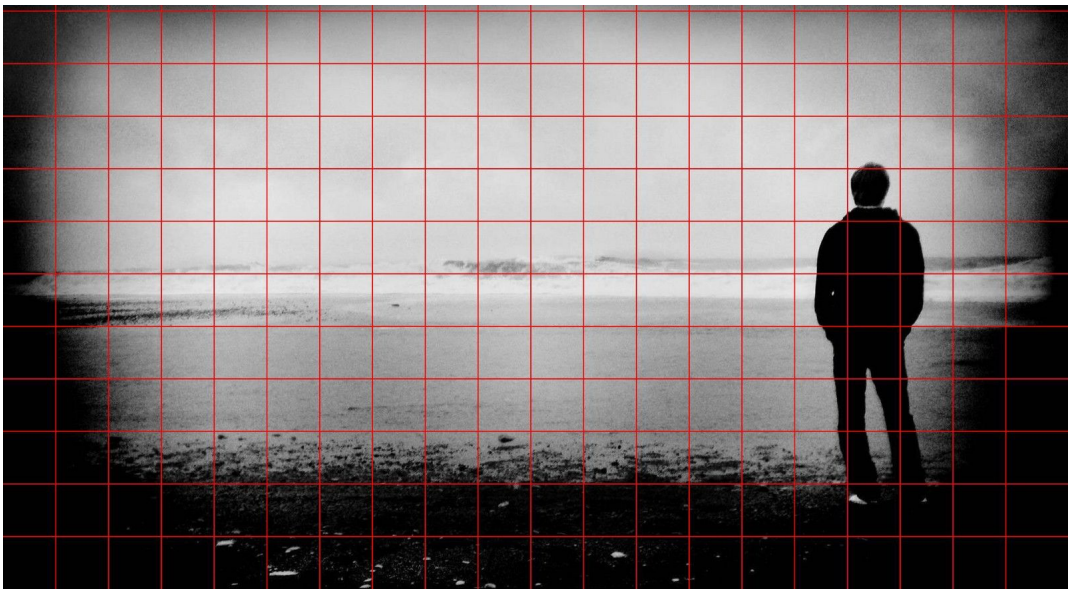
     Y adalah *brightness* dari suatu citra, Cb adalah nilai perbedaan warna biru relatif terhadap warna hijau, dan Cr adalah nilai

perbedaan warna merah relatif terhadap warna hijau. Konversi ini dapat dilakukan dengan melakukan kalkulasi yang sudah terstruktur.

ii. Preproses DCT

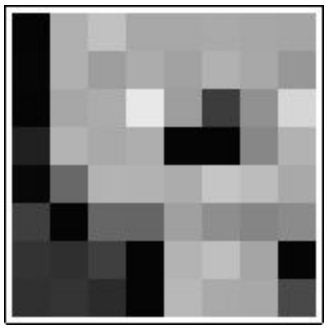
Citra dipartisi per blok dengan ukuran blok adalah 8x8 pixel. Berikut adalah contoh pempartisian gambar berukuran 240x320 pixel.

Gambar 1. Citra sebelum dipartisi



Gambar 2. Citra sesudah dipartisi

Kemudian dimisalkan akan diproses partisi dari baris ke-10, kolom ke-3. Yaitu :



Gambar 3. Partisi citra dari baris ke-10, kolom ke-3.

Dengan matriks (setiap nilai telah dikurangi 128 agar nilai setiap bilangan menjadi signed integer ::

–122	0049	0066	0041	0041	0043	0040	0038
–121	0049	0031	0045	0035	0050	0041	0024
–122	0040	0045	0105	0031	–066	0018	0087
–094	0052	0042	0047	–122	–122	0008	0051
–119	–023	0053	0051	0045	0070	0061	0042
–064	–122	–025	–026	0033	0015	0006	0012
–076	–080	–064	–122	0053	0064	0038	–122
–078	–074	–084	–122	0057	0043	0041	–053

iii. Pemprosesan DCT (Discrete Cosine Transformasi)

Setiap nilai pada matriks di atas kemudian ditransformasi menjadi bentuk gelombang cosinus yang nilainya berada di antara -1 dan 1. Transformasi ini bertujuan menghilangkan frekuensi tinggi yang tidak dapat begitu dilihat oleh mata manusia. Berikut adalah rumus dan hasil yang diperoleh setelah transformasi DCT :

$$f_{x,y} = \frac{1}{4} \sum_{u=0}^{7} \sum_{v=0}^{7} \alpha(u)\alpha(v) F_{u,v} \cos\left[\frac{(2x+1)u\pi}{16}\right] \cos\left[\frac{(2y+1)v\pi}{16}\right]$$

-27.500	-213.468	-149.608	-95.281	-103.750	-46.946	-58.717	27.226
168.229	51.611	-21.544	-239.520	-8.238	-24.495	-52.657	-96.621
-27.198	-31.236	-32.278	173.389	-51.141	-56.942	4.002	49.143
30.184	-43.070	-50.473	67.134	-14.115	11.139	71.010	18.039
19.500	8.460	33.589	-53.113	-36.750	2.918	-5.795	-18.387
-70.593	66.878	47.441	-32.614	-8.195	18.132	-22.994	6.631
12.078	-19.127	6.252	-55.157	85.586	-0.603	8.028	11.212
71.152	-38.373	-75.924	29.294	-16.451	-23.436	-4.213	15.624

iv. Co-Efficient *Quantization*

Pada langkah ini, setiap nilai yang mendekati 0 akan diubah menjadi 0, dan sisanya mengecil mendekati 0. Kuantisasi ini dilakukan dengan mengalkulasi matriks dari hasil langkah sebelumnya dengan rumus dan matriks kuantisasi ini :

$$B_{j,k} = \text{round}\left(\frac{A_{j,k}}{Q_{j,k}}\right)$$

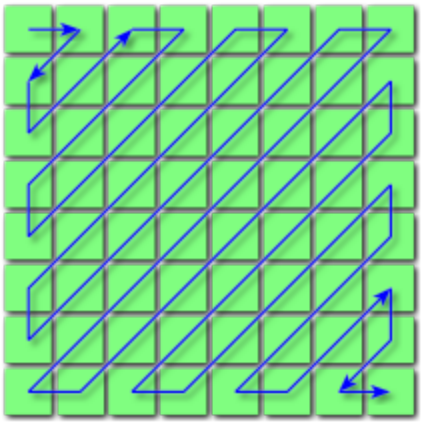
$$Q_{j,k} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Maka hasil yang diperoleh adalah :

-2	-19	-15	-6	-4	-1	-1	0
14	4	-2	-13	0	0	-1	-2
-2	-2	-2	7	-1	-1	0	1
2	-3	-2	2	0	0	1	0
1	0	1	-1	-1	0	0	0
-3	2	1	-1	0	0	0	0
0	0	0	-1	1	0	0	0
1	0	-1	0	0	0	0	0

v. *Lossless Encoding*

Hasil dari kuantisasi di atas kemudian dipetakan menjadi larik 1 dimensi dengan metode zigzag dan metode kompresi Huffman.



Gambar 4. Metode kompresi zigzag

Langkah terakhir ini bertujuan mengelompokkan nilai dengan frekuensi bukan 0 di awal dan frekuensi bukan 0 di akhir. Maka hasil penerapan langkah ini menghasilkan :
-2, -19, 14, -2, 4, -15, -6, -2, -2, 2 1 -3, -2, -13, -4 -1, 0, 7, -3, 0, -3, 0, 2, 1, 2, -1, 0, -1, 0, -1, -1, 0, -1, 1, 0, 1, 0, 0, -1, -1, 0, 0, -2, 1, 1, 0, 0, -1, -1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.

Hasil rentetan bilangan dalam larik 1 dimensi tersebut kemudian dikompres dengan metode huffman. Dan dapat diperoleh hasil kompresi hingga 70% lebih kecil dari ukuran aslinya.

- MP3

MP3 (yang berasal dari MPEG-1 Audio Layer III dan MPEG-2 Audio Layer III) adalah *audio coding format* yand dibuat untuk audio digital. Algoritma MP3 secara mendetil tidak akan dibahas pada artikel ini karena begitu kompleks dan membutuhkan perhitungan, dan kalkulasi yang sangat situasional dan relatif pada jenis audio yang akan dikompresi.

Metode ini mengompresi *file* audio dengan memperkirakan atau menghapus sebagian komponen suara yang dianggap melebihi kemampuan indera pendengaran manusia. Metode ini perseptual, dan menggunakan teknik pemodelan psikoakustik. Metode kompresi MP3

juga seringkali dapat mengompresi ukuran *file* hingga 95% lebih kecil dari ukuran aslinya.

- MPEG-1

  MPEG-1 (Moving Pictures Experts Group) adalah metode kompresi video yang ditemukan pada tahun 1988. Terdapat banyak metode kompresi yang diciptakan oleh MPEG. Namun pada artikel ini hanya dibahas metode MPEG-1 secara garis besar. Video adalah rentetan gambar yang disusun dan berganti sekian kali setiap detik. Metode kompresi MPEG secara umum memanfaatkan kesamaan antara perubahan satu gambar ke gambar yang lain agar setiap bagian yang hampir sama tidak di-*update*. Sehingga hanya perlu disimpan data dengan jumlah yang lebih sedikit

  Metode ini dibagi menjadi 5 bagian yaitu :

  i. Sistem

     Langkah pertama ini menentukan metode penyimpanan, sinkronisasi video, audio, data lainnya secara bersamaan.

  ii. Video

     Kompresi video ini memiliki beberapa jenis tipe *frame*.

     1. I-Frame (Intra-Frame)

        Frame ini adalah frame yang metode kompresinya sama persis dengan tahap kompresi citra JPEG yang sudah dibahas di poin pertama di atas

     2. P-Frame (Prediction-Frame)

        Frame ini memanfaatkan pengulangan gambar yang digunakan pada rentetan gambar dan hanya mencatat perbedaannya.

     3. B-Frame (Bidirectional-Frame)

        Frame ini mirip dengan P-Frame, hanya saja dapat membuat prediksi dari masa lalu (gambar sebelumnya) dan masa depan (gambar tebakan berikutnya).

     Setelah framing, kemudian dilakukan *lossless compression* untuk menangani perubahan rentetan gambar.

  iii. Audio

Kompresi ini memiliki 3 lapisan. Lapisan yang umum dipakai adalah lapisan ketiga dan, kompresi lapisan ketiga ini menggunakan metode MP3 yang sudah dibahas pada poin kedua di atas.

iv. *Compliance Testing* dan *Software Simulation*

Dua metode terakhir ini bertujuan untuk mengevaluasi dan menguji hasil kompresi yang telah dikalkulasi agar tetap terbentuk hasil kompresi yang *acceptable*.

B. Lossless Compression

Lossless compression adalah sebuah teknik untuk mengurangi data yang disimpan dengan tidak ada informasi yang hilang. Data hasil kompresi harus dapat direkonstruksi ulang menjadi sama seperti data awal. *Lossless compression* digunakan pada berbagai macam tipe *file*, seperti untuk kompresi data audio, gambar, maupun video, maupun f*ile* general lainnya. Secara umum, proses dalam lossless compression dibagi menjadi dua, yaitu menghasilkan model statistik untuk memperkecil ukuran *file*, kemudian menggunakan model statistik tersebut untuk memetakan data masukan menjadi bentuk yang telah dikompresi. Tidak ada satu teknik yang dapat mengompresi seluruh tipe data secara efisien. Oleh karena itu, setiap teknik kompresi dibuat untuk sebuah tipe data yang spesifik. Berikut beberapa contoh teknik kompresi data:

- Run-Length Encoding (RLE)

  Teknik Run-Length Encoding merupakan salah satu teknik paling sederhana untuk kompresi secara *lossless*. Hampir seluruh *file bitmap* dapat dikompresi menggunakan teknik RLE. Teknik RLE tidak mempedulikan informasi dari data yang ingin dikompresi, namun isi dari data tersebut memengaruhi seberapa efisien teknik RLE dapat bekerja. Contoh sederhana dari kompresi secara RLE adalah sebagai berikut :

  Sebelum kompresi

AAAAAABBBBB

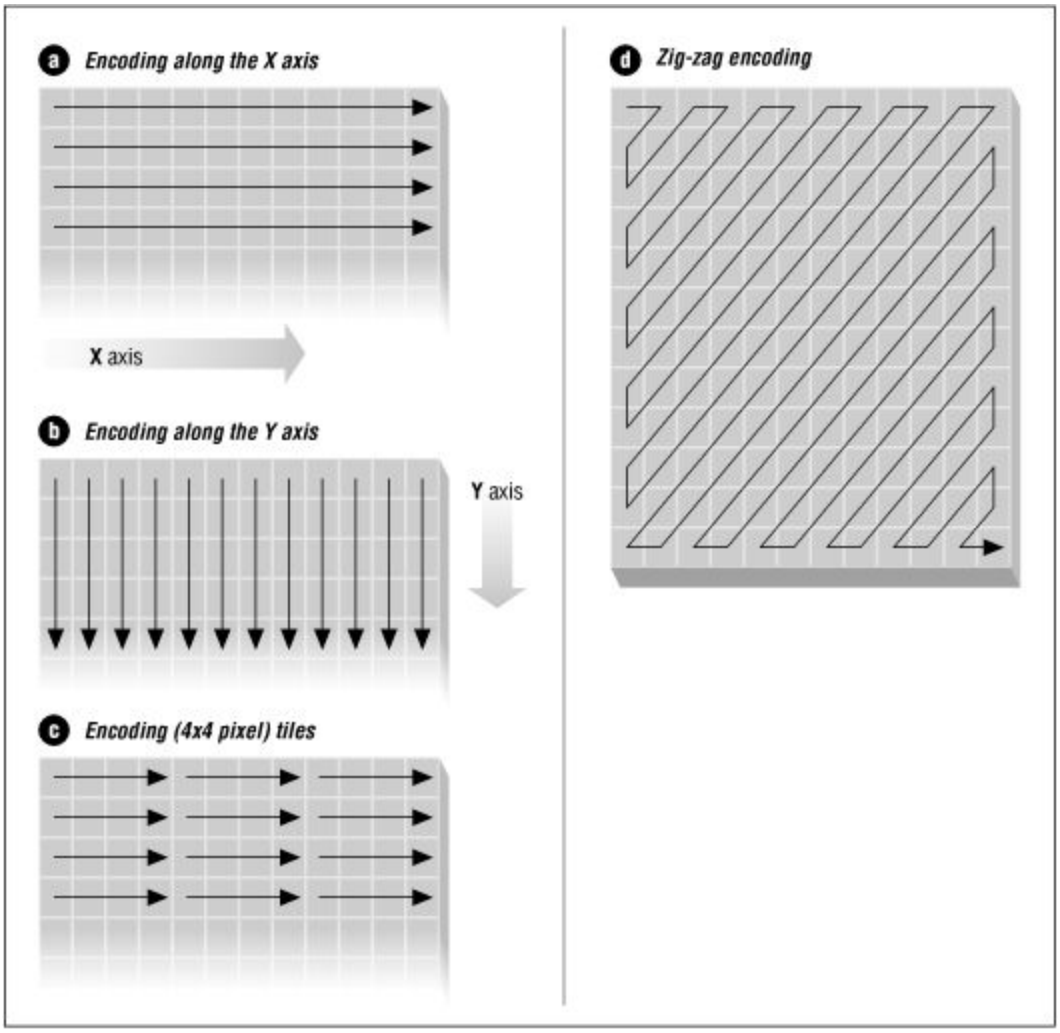
  Setelah kompresi

6A5B

  Pada contoh di atas, jumlah karakter setelah dikompresi menjadi berkurang, dari awalnya 11 karakter menjadi 4 karakter. Kompresi RLE menyimpan jumlah karakter yang ada secara kontigu, dan karakter yang ada tersebut. Data yang ingin dikompresi sangat berpengaruh terhadap seberapa efisien RLE dapat mengkompresinya.

Keunggulan dari kompresi RLE adalah kompresi model ini sederhana dan mudah untuk diimplementasikan. Namun, RLE tidak bekerja dengan baik terhadap semua masukan. Bahkan, hasil kompresi dapat lebih besar dibandingkan ukuran aslinya. RLE tidak bekerja dengan baik untuk mengompresi hasil foto, karena hasil dari foto memiliki warna yang sangat beragam dan jarang ditemui hasil foto yang memiliki warna yang sama secara kontigu yang cukup panjang.

Terdapat beberapa variasi untuk kompresi secara RLE. Umumnya, untuk mengompresi sebuah gambar, dilakukan dengan menganggap data gambar tersebut sebagai sebuah data sekuensial yang mengalir secara satu dimensi. Untuk pemrosesan secara sekuensial, dapat dilakukan dari ujung kiri atas ke kanan, dilakukan terus hingga ke ujung kanan bawah (gambar 1 bagian a), ataupun dari ujung kiri atas ke bawah hingga ke ujung kanan bawah (gambar 1 bagian b). Terdapat juga variasi RLE secara non-sekuensial, yaitu dengan membagi-bagi setiap bagian menjadi kotak-kotak kecil (gambar 1 bagian c) dan zig-zag (gambar 1 bagian d). Kompresi RLE dapat divariasikan dengan berbagai macam bentuk lainnya, namun jarang dilakukan variasi lain tersebut.



Gambar 1. Variasi Teknik Run-Length Encoding (RLE)

- Portable Network Graphics (PNG)

PNG adalah sebuah teknik kompresi dari *file* gambar yang masih disimpan secara dot matriks secara *lossless*. Kompresi pada *file* PNG dibagi menjadi dua tahap, yaitu *filtering* (prediksi), dan kompresi (*deflate*). Tahap *filtering* pada *file* PNG yaitu mentransformasi data yang awalnya adalah nilai dari setiap piksel menjadi perbedaan nilai dengan piksel yang berada di dekatnya. Untuk setiap garis-pengecekan, sebuah piksel dipetakan dengan relasi terhadap atas, kiri, dan diagonal kiri atas.

PNG memiliki lima buah teknik untuk filtering, yaitu tidak difilter, perbedaan dengan piksel kiri, perbedaan X dengan piksel atas, perbedaan dengan (piksel kiri + piksel atas)/2, dan prediksi (fungsi linear dari piksel di kiri, atas, dan diagonal kiri atas. Setiap teknik memiliki kasus terbaik dan kasus

terburuknya masing-masing. Setelah dilakukan *filtering*, dilakukan kompresi, yaitu dengan algoritma *deflate.*

Algoritma *deflate* merupakan kombinasi algoritma LZ277 dengan pengkodean Huffman. Algoritma ini mengecek kecocokan panjang antara 3 hingga 258 simbol, sehingga dapat membuat rasio maksimum kompresi data adalah 1032:1. Jika kecocokan kurang dari 3 simbol, dibutuhkan informasi tambahan untuk merepresentasikan piksel. Kompresi secara PNG sangat bergantung terhadap kecocokan pola pada gambar. Jika kecocokan ternyata tinggi, maka *file* PNG dapat dikompresi secara efisien.

Gambar 2. Perbandingan JPG dan PNG

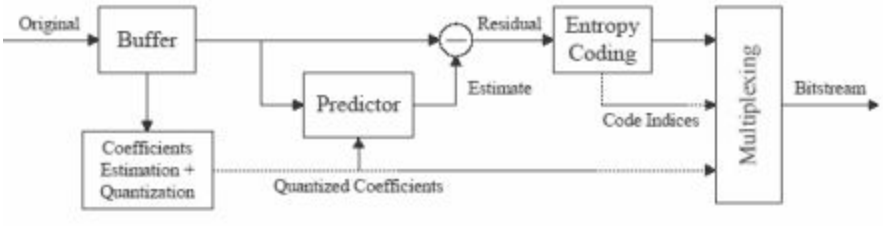
- MPEG-4 Audio Lossless Coding (MPEG-4 ALS)

  MPEG-4 ALS merupakan sebuah pengembangan dari MPEG-4 yang dapat mengompresi secara *lossless*. Teknik kompresi ini difinalisasi pada Desember 2005. MPEG-4 ALS mengompresi data berdasarkan prediksi linear secara adaptif-maju, yang dapat mengompresi secara cepat. Proses ini dilakukan Fitur tambahan seperti prediksi jangka panjang, kode *multichannel*, dan kompresi bilangan real juga dapat diterapkan pada MPEG-4 ALS. MPEG-4 ALS dapat mengompresi secara efisien dalam hal rasio kompresi dan juga kecepatannya.

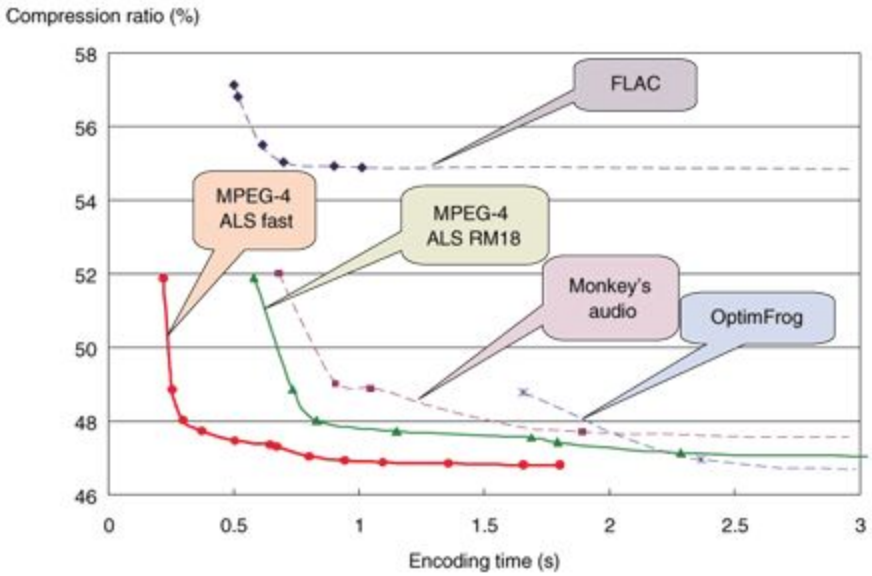
  Proses *encode* menggunakan teknik MPEG-4 ALS terbagi menjadi beberapa bagian, seperti yang terlihat pada gambar 3, yaitu :

  - ➢ *Buffer* : menyimpan frame audio yang terdiri dari satu atau lebih sampel PCM yang merepresentasikan sinyal untuk setiap channel audio
  - ➢ *Coefficients Estimator and Quantizer* : memperkirakan dan mengkuantisasi prediksi yang optimal untuk setiap blok

- ➢ *Predictor* : Mengkalkulasi prediksi residu dari sinyal yang asli dan koefisien kuantifikasi
- ➢ *Entropy Coding* : Mengkodekan resudi menggunakan Golomb-Rice ataupun algoritma entropi lainnya
- ➢ *Multiplexer* : Mengombinasikan residu hasil *entropy coding*, indeks kode, koefisien prediksi, dan kode CRC



Gambar 3. Proses *Encode* MPEG-4 ALS



Gambar 4. Perbandingan Performa Proses Kompresi Audio

Daftar Pustaka

- https://techterms.com/definition/lossy
- https://medium.com/@danojadias/jpeg-compression-algorithm-969af03773da
- https://www.tutorialspoint.com/dip/introduction_to_jpeg_compression.htm
- https://cs.stanford.edu/people/eroberts/courses/soco/projects/data-compression/lossy/mp3/index.htm
- https://en.wikipedia.org/wiki/MPEG-1
- https://cs.stanford.edu/people/eroberts/courses/soco/projects/data-compression/lossless/index.htm

- https://www.w3.org/Graphics/PNG/
- https://medium.com/@duhroach/how-png-works-f1174e3cc7b7
- https://www.fileformat.info/mirror/egff/ch09_03.htm
- https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr200712sp3.html