



PAPER – OPEN ACCESS

Perbandingan Algoritma Huffman Dan Run Length Encoding Untuk Kompresi File Audio

Author : Helbert Sinaga

DOI : 10.32734/st.v1i1.183

Electronic ISSN : 2654-7093

Print ISSN : 2654-7085

Volume 1 Issue 1 – 2018 TALENTA Conference Series: Science & Technology (ST)



This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/).

Published under licence by TALENTA Publisher, Universitas Sumatera Utara



Perbandingan Algoritma Huffmandan Run Length Encoding Untuk Kompresi File Audio

Helbert Sinaga^a, Poltak Sihombing^a, Handrizal^a

Program Studi SI Ilmu Komputer, Fasilkom-TI Universitas Sumatera Utara Medan 20155, Indonesia

handrizal_tanjung@yahoo.com

Abstrak

Penelitian ini dilakukan untuk menganalisis perbandingan hasil kompresi dan dekompresi file audio *.mp3 dan *.wav. Kompresi dilakukan dengan mengurangi jumlah bit yang diperlukan untuk menyimpan atau mengirim file tersebut. Pada penelitian ini penulis menggunakan algoritma Huffman dan Run Length Encoding yang merupakan salah satu teknik kompresi yang bersifat lossless. Algoritma Huffman memiliki tiga tahapan untuk mengkompres data, yaitu pembentukan pohon, encoding dan decoding dan bekerja berdasarkan karakter per karakter. Sedangkan teknik run length ini bekerja berdasarkan sederetan karakter yang berurutan, yaitu hanya memindahkan pengulangan byte yang sama berturut-turut secara terus-menerus. Implementasi algoritma Huffman dan Run Length Encoding ini bertujuan untuk mengkompresi file audio *.mp3 dan *.wav sehingga ukuran file hasil kompresi lebih kecil dibandingkan file asli dimana parameter yang digunakan untuk mengukur kinerja algoritma ini adalah rasio kompresi, kompleksitas yang dihasilkan. Rasio kompresi file audio *.mp3 menggunakan Algoritma Huffman memiliki rata-rata 1.204% sedangkan RLE -94.44%, dan rasio kompresi file audio *.wav memiliki rata-rata 28.954 % sedangkan RLE -45.91%.

Kata Kunci: kompresi; dekompresi; mp3; wav; Huffman; Run Length Encoding; lossless

1. Pendahuluan

Dalam perkembangan teknologi saat ini, komputer digunakan untuk membantu dan mempercepat kinerja manusia. Dalam hal komunikasi data, pesan yang dikirim seringkali ukurannya sangat besar sehingga waktu pengirimannya lama. Begitu juga dalam penyimpanan data, arsip atau file yang berukuran besar membutuhkan ruang penyimpanan yang besar. Kedua masalah tersebut dapat diatasi dengan mengkodekan pesan atau isi arsip sesingkat mungkin, sehingga waktu pengiriman data juga relatif cepat, dan ruang penyimpanan yang dibutuhkan juga sedikit. Cara pengkodean seperti ini disebut pemampatan atau kompresi data.

Pada umumnya file audio memiliki ukuran yang lebih besar dibandingkan file teks, file gambar sehingga untuk mengurangi ukurannya dapat dilakukan kompresi. Seperti teknik kompresi pada umumnya ada dua jenis kompresi data yaitu lossy dan lossless. Kompresi lossless adalah kompresi file, file dikembalikan seperti data asli tanpa mengalami perubahan atau data tidak ada yang hilang. Maka salah satu solusi untuk masalah di atas adalah melakukan kompresi data sebelum ditransmisikan dan kemudian si penerima akan melakukan dekompresi dengan mengubahnya kembali secara sempurna tanpa terjadi kehilangan informasi dari data asli.

Algoritma Huffman merupakan algoritma yang terkenal untuk kompresi data atau file bersifat lossless. Algoritma Huffman memiliki tiga tahapan untuk mengkompres data, yaitu pembentukan pohon, encoding dan decoding. Berbeda dengan algoritma Huffman yang bekerja berdasarkan karakter per karakter, tetapi algoritma Run Length Encoding

bekerja sederetan karakter yang berurutan. Algoritma Run Length Encoding adalah algoritma kompresi data bersifat lossless juga.

Dalam analisis perbandingan kompresi file audio menggunakan algoritma Huffman dan Run Length Encoding yang mana paling ideal sehingga melakukan penelitian yang berjudul: —Analisis Perbandingan Algoritma Huffman Dan Run Length Encoding Pada Kompresi File Audiol.

Jika dilakukan perhitungan nilai kriteria ukuran pemusatan, lokasi, dan dispersi pada data berkelompok dalam tabel distribusi frekuensi akan terjadi perbedaan perhitungan dengan data tunggalnya (sebelum dikelompokkan). Perbedaan demikian dikatakan sebagai selisih akibat pengelompokkan (grouping error) [2]. Melalui permasalahan tersebut, dalam penelitian ini dilakukan analisa perbandingan keakuratan dari model tabel distribusi frekuensi berkelompok antara penggunaan metode sturges dan metode scott terhadap nilai selisih pengelompokkan yang berdasarkan nilai kriteria rata-rata, median, kuartil 1, kuartil 3, standar deviasi, dan varians data.

2. Kompresi

Kompresi data (data compression) bertujuan meminimalkan kebutuhan memori untuk merepresentasikan data digital. Prinsip umum yang digunakan pada proses kompresi adalah mengurangi duplikasi data sehingga memori untuk merepresentasikan menjadi lebih sedikit dari pada representasi data semula [1].

2.1. Algoritma Huffman

Algoritma Huffman adalah salah satu algoritma kompresi tertua yang disusun oleh David Huffman pada tahun 1952. Algoritma tersebut digunakan untuk membuat kompresi jenis lossless compression, yaitu pemampatan data dimana tidak satu byte pun hilang sehingga data tersebut utuh dan disimpan sesuai dengan aslinya. Pada dasarnya, algoritma Huffman ini bekerja seperti mesin sandi morse, dia membentuk suatu kode dari suatu karakter. Sehingga karakter tersebut memiliki rangkaian bit yang lebih pendek dibandingkan sebelumnya [2].

Adapun tahapan dari algoritma Huffman adalah yang pertama membuat sebuah pohon berupa simpul daun dan anak-anaknya yang memiliki probabilitas dari seringnya munculnya karakter. Proses kedua adalah proses encoding, dari pohon tersebut, tiap karakternya akan memiliki identitas berupa bilangan biner untuk disimpan pada memori. Proses pembentukan dari karakter menjadi biner inilah yang disebut dengan proses encoding. Untuk lebih jelasnya bagaimana encoding itu akan diperjelas ketika pohon telah dibuat. Proses ketiga yaitu proses decoding adalah proses kebalikan dari encoding, yaitu mengembalikan dari angka-angka biner yang pendek diubah lagi menjadi karakter yang panjang lagi dan tanpa ada kehilangan data dari data asli.

2.2 Algoritma Run Length Encoding

Teknik run length ini bekerja berdasarkan sederetan karakter yang berurutan. Run Length Encoding adalah suatu algoritma kompresi data yang bersifat lossless. Algoritma ini mungkin merupakan algoritma yang mudah untuk dipahami dan diterapkan untuk kompresi. Metode kompresi ini sangat sederhana, yaitu hanya memindahkan pengulangan byte yang sama berturut-turut secara terus-menerus [3].

2.3 Rasio Kompresi

tingkat pengurangan data yang dicapai sebagai hasil dari proses kompresi disebut rasio kompresi. Rasio ini merupakan perbandingan antara panjang data asli dengan panjang data yang sudah dikompresi, seperti dituliskan dalam persamaan berikut:

$$\text{RasioKompresi} = \frac{\text{UkuranFileAsli}}{\text{UkuranFileTerkompresi}} \quad (1)$$

Jika dinyatakan dalam persentase maka dituliskan dalam persamaan berikut:

$$P = \left(1 - \frac{\text{UkuranFileTerkompresi}}{\text{UkuranFileAsli}}\right) \times 100\% \tag{2}$$

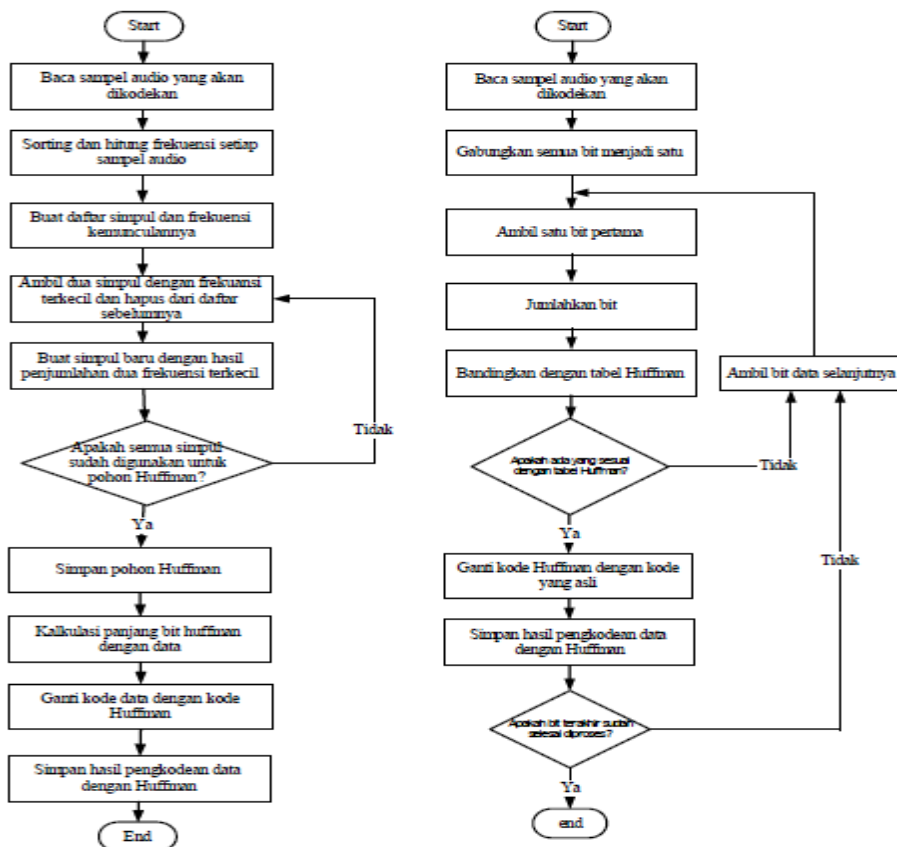
Yang berarti ukuran file berkurang sebesar P (dalam persentase) dari ukuran semula. Semakin tinggi rasio tingkat suatu teknik kompresi data maka semakin efektif teknik kompresi tersebut.

2.4 Kompleksitas Waktu

Perhitungan Kompleksitas waktu merupakan waktu yang dibutuhkan oleh sebuah sistem untuk menginput file audio yang akan dikompresi sampai dengan selesainya proses kompresi dan dekomposisi.

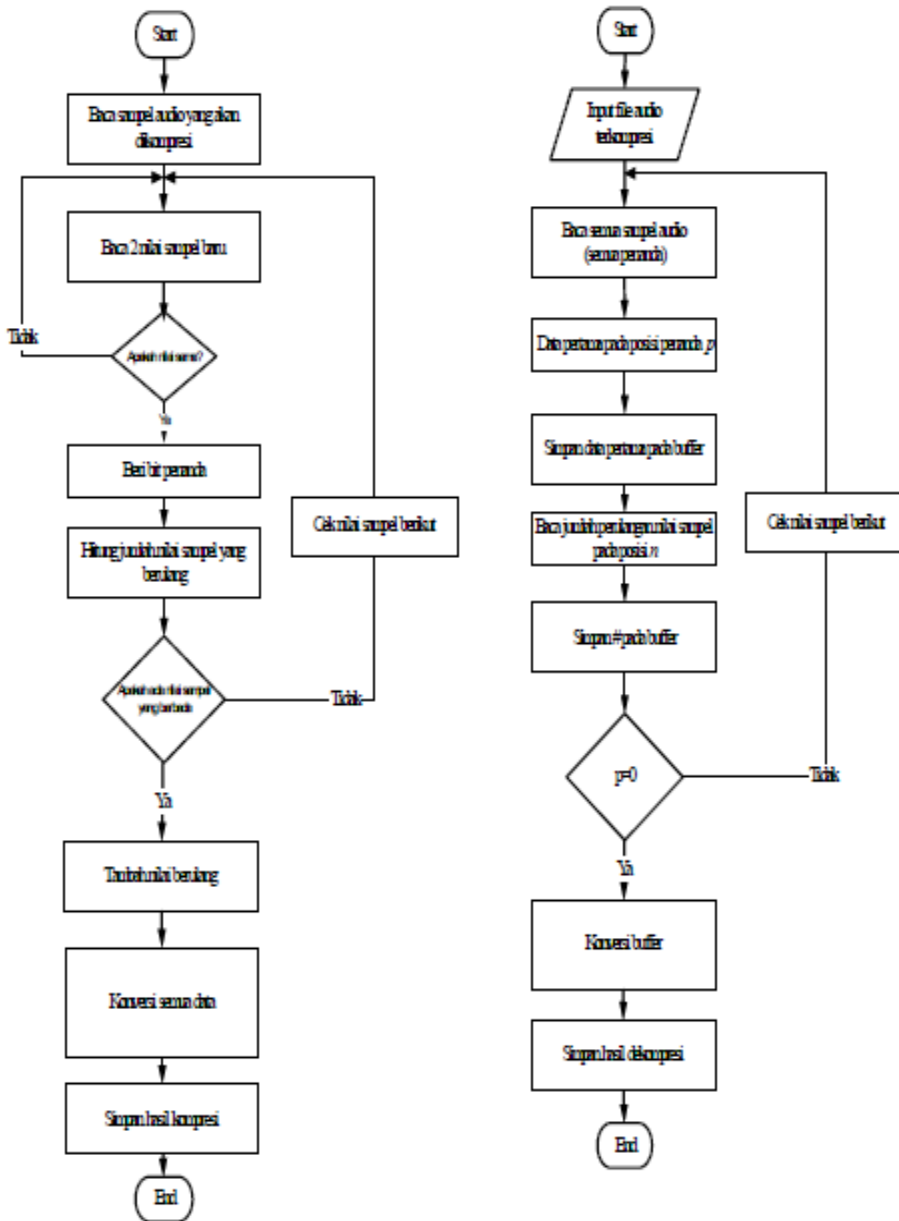
3. Perancangan Sistem

Perancangan sistem dilakukan untuk memberikan gambaran secara umum terhadap implementasi yang akan dibuat. Berikut adalah langkah kerja dari kompresi dan dekomposisi file audio menggunakan algoritma Huffman yang digambarkan dalam bentuk flowchart terlihat pada Gambar 1.



Gambar 1: Flowchart Kompresidan Dekompresi file audio menggunakan algoritma Huffman

Berikut adalah langkah kerja dari kompresi dan dekomposisi file audio menggunakan algoritma Run Length Encoding yang digambarkan dalam bentuk flowchart terlihat pada Gambar 2 berikut.



Gambar 2: Flowchart Kompresi dan Dekompresi file audio menggunakan algoritma Run Length Encoding

4. Perbandingan Kompresi

Setelah melakukan perancangan maka selanjutnya yang harus dilakukan adalah membandingkan kompresi dua algoritma yaitu algoritma Huffman dan Run Length Encoding.

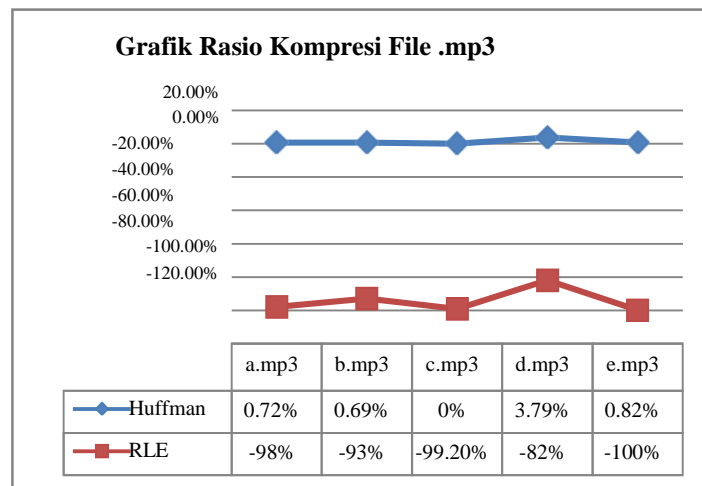
4.1. Perbandingan Rasio Kompresi

Tabel1 berikut merupakan hasil dari lima percobaan proses kompresi file audio berformat *.mp3 menggunakan algoritma Huffman dan Run Length Encoding.

Tabel 1: Perbandingan Kompresi File Audio (*.mp3) menggunakan Algoritma Huffman dan RLE

No	Nama File	Ukuran Asli	Metode			Rasio (%)	
			Huffman	RLE	Huffman	RLE	
1	a.mp3	1.39 Mb	1.38 Mb	2.76 Mb	0.72 %	-98 %	
2	b.mp3	144 Kb	143 Kb	278 Kb	0.69 %	-93 %	
3	c.mp3	1.26 Mb	1.26 Mb	2.51 Mb	0 %	-99.2 %	
4	d.mp3	1.04 Mb	989 Kb	1.80 Mb	3.79 %	-82 %	
5	e.mp3	1.22 Mb	1.21 Mb	2.44 Mb	0.82 %	-100 %	
Rata-rata Rasio Kompresi					1.204 %	-94.44 %	

Dari percobaan pada tabel 1 dapat dilihat untuk rata-rata rasio kompresi algoritma Huffman pada file *.mp3 adalah 1.204 % sedangkan algoritma RLE adalah -94.44. Untuk mengetahui lebih jelas perbandingan kedua algoritma dalam rasio kompresi dapat dilihat pada gambar 3 berikut.



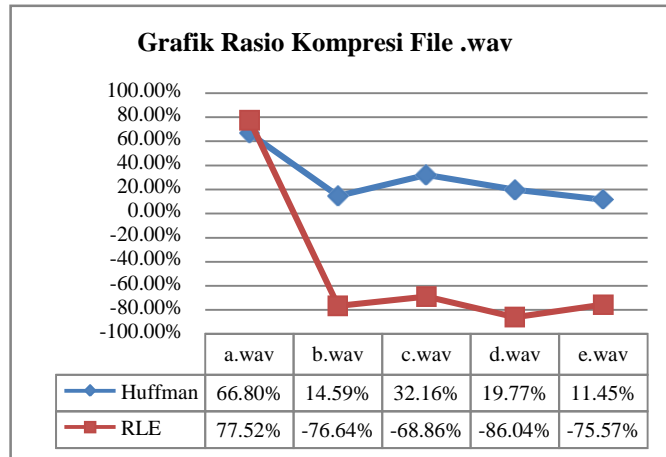
Gambar 3: Grafik Rasio Kompresi File *.mp3.

Tabel 2 berikut merupakan hasil dari lima percobaan proses kompresi file audio berformat *.wav menggunakan algoritma Huffman dan Run Length Encoding.

Tabel 2: Perbandingan Kompresi File Audio (*.wav) menggunakan Algoritma Huffman dan RLE

No	Nama File	Ukuran Asli	Metode			Rasio (%)	
			Huffman	RLE	Huffman	RLE	
1	a.wav	21.8 Kb	6.8 Kb	16.9 Kb	66.8 %	77.52 %	
2	b.wav	137 Kb	117 Kb	242 Kb	14.59 %	-76.64 %	
3	c.wav	106 Kb	71.9 Kb	179 Kb	32.16 %	-68.86 %	
4	d.wav	172 Kb	138 Kb	320 Kb	19.77 %	-86.04 %	
5	e.wav	131 Kb	116 Kb	230 Kb	11.45 %	-75.57 %	
Rata-rata Rasio Kompresi					28.954 %	-45.91 %	

Dari percobaan pada tabel 2 dapat dilihat rata-rata rasio kompresi algoritma Huffman pada file *.wav adalah 28.954 % sedangkan algoritma RLE adalah -54.91 %. Untuk mengetahui lebih jelas perbandingan kedua algoritma dalam rasio kompresi dapat dilihat pada gambar 4 berikut.



Gambar 4: Grafik Rasio Kompresi File *.wav.

4.2 Perhitungan Kompleksitas Waktu

4.2.1 Algoritma Huffman

$$T(n) = (C_1 + C_2 + C_6 + 8C_9) n^0 + (7C_3 + C_4 + 2C_5 + 3C_7 + 8C_8 + 9C_{10} + C_{11} + C_{12}) n^1 + (C_3 + C_{13}) n^2$$

$$= \zeta n \text{ Log } n$$

4.2.2 Algoritma Run Length Encoding

$$T(n) = (C_1 + C_2 + 8C_6) n^0 + (11C_3 + 103C_4 + 26C_5 + 8C_6 + 8C_7 + 4C_8 + 9C_9) n^1$$

$$= \zeta (n)$$

5. Kesimpulan

Percobaan kompresi file yang dilakukan sebanyak total 10 kali, menghasilkan kesimpulan sebagai berikut:

1. Pada file audio berformat *.wav dan *.mp3, algoritma Huffman dapat menghasilkan rasio kompresi dengan persentase yang lebih kecil dibandingkan dengan algoritma Run Length Encoding, karena rata-rata rasio kompresi nya adalah sebagai berikut.
 - a. Rata-rata rasio kompresi file *.mp3 dengan lima kali percobaan menggunakan algoritma Huffman adalah 1.204% sedangkan menggunakan algoritma RLE adalah -94.44%, dan
 - b. Rata-rata rasio kompresi file *.wav dengan lima kali percobaan menggunakan algoritma Huffman adalah 28.954% sedangkan menggunakan algoritma RLE adalah -45.91%.

Sehingga dapat disimpulkan bahwa algoritma Huffman lebih baik dari pada algoritma RLE dalam kompresi file audio *.mp3 maupun *.wav.

2. Penghitungan kompleksitas waktu algoritma Huffman adalah $T(n) = \zeta n \text{ Log } n$ sedangkan algoritma Run Length Encoding adalah $T(n) = \zeta (n)$.

Referensi

- [1] Munir, Rinaldi. 2004. Pengolahan Citra Digital dengan Pendekatan Algoritmik. Bandung: Informatika.
- [2] Rajagukguk. M. D. 2014. Analisis Perbandingan Algoritma Huffman Dengan Algoritma (Lempel-Zip- Welch) Pada Kompresi Gambar Menggunakan Metode Exponensial. Jurnal. STMIK Budidarma. Medan.
- [3] Rahandi, Aditya. 2009. Analisis dan Implementasi Kompresi File Audio Dengan Menggunakan Algoritma Run Length Encoding (RLE). Jurnal. Universitas Sumatera Utara.