*J-INTECH (Journal of Information and Technology)*
**Terakreditasi Kemendikbud SK No. 204/E/KPT/2022**
**E-ISSN: 2580-720X || P-ISSN: 2303-1425**

# Komparasi Performa Algoritma Kompresi *Data Lossless* Menggunakan Rasio Kompresi Dan Penghematan Ruang

Aswar Hanif [1*], Endang Wahyudi[2], Harna Adianto[3], Lilik Martanto[4]

[1,2,3,4]*Universitas Bina Sarana Informatika, Fakultas Teknik & Informatika, Jakarta, Indonesia*

***Informasi Artikel***

Diterima: 22-05-2022
Direvisi: 23-06-2023
Diterbitkan: 12-07-2023

***Kata Kunci***
*Kompresi; Lossless; Rasio kompresi; Penghematan ruang*

***Email Korespondensi:**
*aswar.awf@bsi.ac.id*

***Abstrak***
Pertumbuhan data merupakan tantangan besar. Tujuan dari kompresi data adalah untuk mengurangi jumlah bit yang dibutuhkan untuk merepresentasikan informasi yang berguna. Kompresi data dapat digunakan untuk meningkatkan efisiensi penyimpanan, transmisi, dan perlindungan data. Algoritma lossless dapat dengan persis merekonstruksi data asli dari data terkompresi. Kompresi lossless sering digunakan untuk data yang perlu disimpan atau dikirim secara akurat. Beberapa metode dan algoritma kompresi lossless diantaranya seperti Lempel–Ziv–Markov chain algorithm (LZMA), Prediction by partial matching (PPM), Burrows-Wheeler block sorting text compression algorithm and Huffman coding (BZip2), dan Deflate. Meskipun semua sistem kompresi didasarkan pada prinsip yang sama, tetap seharusnya ada perbedaan performa. Karena itu dibutuhkan sebuah panduan umum untuk membantu menentukan pilihan algoritma kompresi data yang paling tepat untuk digunakan. Penelitian ini bertujuan untuk menentukan algoritma kompresi data yang memiliki performa terbaik, berdasarkan komparasi menggunakan nilai Rasio Kompresi dan Penghematan Ruang. Tahapan penelitian dimulai dengan penentuan algoritma kompresi yang digunakan, persiapan data, pengujian performa, untuk kemudian dilakukan pembahasan dan ditarik kesimpulan. Hasil penelitian menunjukkan bahwa rasio kompresi dan penghematan ruang yang bisa dicapai secara spesifik akan bergantung pada data yang digunakan. Meskipun rata-rata nilai performa kompresi tidak terlalu jauh, tapi secara umum LZMA2 menunjukkan hasil terbaik dengan rasio kompresi 1,457 dan penghematan ruang 15,00%. Hasil pengujian ini diharapkan dapat digunakan sebagai gambaran umum dalam membantu memilih algoritma kompresi data *lossless*.

***Abstract***
*Data growth is a sizeable challenge. The goal of data compression is to reduce the size of data needed to still represent useful information. Data compression can be used to increase the efficiency of data storage, transmission and protection. Lossless algorithms can precisely reconstruct the original data from the compressed data. Lossless compression is often used for data that needs to be stored or transmitted accurately. Several lossless compression methods and algorithms include the Lempel–Ziv–Markov chain algorithm (LZMA), Prediction by partial matching (PPM), Burrows-Wheeler block sorting text compression algorithm and Huffman coding (BZip2), and Deflate.*

*Even though all compression systems are based on the same principles, there should still be differences in performance. Because of that, a general guide is needed to help determine the most appropriate data compression algorithm to use. This study aims to determine the data compression algorithm that has the best performance, based on a comparison using the Compression Ratio and Space Saving values. The research phase begins with determining the compression algorithm used, data preparation, performance testing, to then be discussed and conclusions drawn. The results show that the compression ratio and space savings that can be achieved specifically will depend on the data used. Although the range of average values of compression performance is not that big, in general LZMA2 shows the best results with a compression ratio of 1.457 and a space saving of 15.00%. Hopefully, the results of this test can be used as an overview in helping to choose a lossless data compression algorithm.*

## 1. Pendahuluan

Data adalah salah satu aset paling berharga di dunia saat ini. Jumlah data yang dihasilkan di dunia tumbuh secara pesat. Pada tahun 2020, dunia menghasilkan 64 miliar terabyte data, dan diperkirakan pada tahun 2025, dunia akan menghasilkan 175 miliar terabyte data (Taylor, 2022). Pertumbuhan data ini didorong oleh sejumlah faktor, termasuk meningkatnya penggunaan perangkat yang terhubung, pertumbuhan media sosial, dan munculnya komputasi awan. Namun, hanya sebagian kecil dari data ini yang tersimpan, karena hanya dua persen dari data yang dihasilkan dan digunakan pada tahun 2020 tersimpan hingga tahun 2021. Pertumbuhan data merupakan tantangan besar, tetapi juga merupakan peluang besar. Bisnis, organisasi, dan individu dapat memanfaatkan kekuatan data untuk memiliki posisi yang baik di masa depan.

Kompresi data adalah proses mengurangi ukuran data dengan tetap menjaga integritas dan kualitasnya. Tujuan dari kompresi data adalah untuk mengurangi jumlah bit yang dibutuhkan untuk merepresentasikan informasi yang berguna (Yang et al., 2023). Ini adalah alat vital untuk mengelola volume data yang terus berkembang di dunia saat ini. Kita menggunakan kompresi data untuk menyimpan informasi atau file dengan cara yang aman (Haque & Huda, 2017). Kompresi data dapat digunakan untuk meningkatkan efisiensi penyimpanan, transmisi, dan perlindungan data. Ini adalah bagian penting dari dunia digital modern. Pentingnya kompresi data di masa mendatang hanya akan bertambah seiring jumlah data yang dihasilkan dan disimpan terus meningkat.

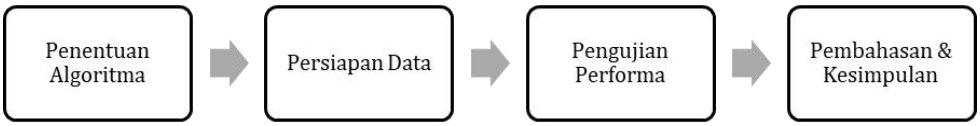
Ada dua jenis utama kompresi data, yaitu lossless dan lossy. Algoritma lossless dapat dengan persis merekonstruksi data asli dari data terkompresi, sedangkan algoritma lossy, hanya dapat merekonstruksi sebuah perkiraan dari data asli (Hosseini, 2012). Kompresi lossless sering digunakan untuk data yang perlu disimpan atau dikirim secara akurat. Ada banyak metode dan algoritma kompresi lossless yang berbeda, diantaranya seperti Lempel–Ziv–Markov chain algorithm (LZMA), Prediction by partial matching (PPM), Burrows-Wheeler block sorting text compression algorithm and Huffman coding (BZip2), dan Deflate. Meskipun semua sistem kompresi didasarkan pada prinsip yang sama, yaitu meringkas data dengan menghilangkan redundansi (Usama et al., 2021), tetap seharusnya ada perbedaan performa yang disebabkan perbedaan cara mereduksi ukuran data yang digunakan oleh setiap algoritma. Karena itu dibutuhkan sebuah panduan umum untuk mengetahui performa algoritma kompresi data, untuk menentukan pilihan yang terbaik.

Performa algoritma kompresi data mengukur sebaik apa sebuah algoritma dapat mengurangi ukuran data tanpa menghilangkan integritasnya. Ini berperan penting dalam mengurangi kebutuhan ruang pada media penyimpanan, dan mengurangi kebutuhan bandwidth saat data itu dikirim melalui jaringan. Kinerja algoritma kompresi data dapat dianalisis dalam beberapa aspek. Hasil penelitian terdahulu yang

membandingkan performa algoritma kompresi data menggunakan nilai Rasio Kompresi dan Kecepatan Kompresi-Dekompresi, menemukan bahwa tidak ada algoritma kompresi memiliki performa optimal untuk semua tipe data (Gupta et al., 2017). Penelitian ini bertujuan untuk menentukan algoritma kompresi data yang memiliki performa terbaik, dengan melakukan perbandingan terhadap performa beberapa algoritma kompresi lossless menggunakan nilai Rasio Kompresi dan Penghematan Ruang. Rasio kompresi merupakan aspek yang paling umum digunakan untuk mengukur performa kompresi data. Rasio kompresi yang lebih tinggi menunjukkan bahwa algoritma kompresi lebih efektif dalam mengurangi ukuran data. Penghematan ruang menunjukkan pengurangan ukuran data dibandingkan data yang tidak terkompresi. Aspek ini menunjukkan hasil performa kompresi dalam mengurangi ukuran data. Diharapkan perbandingan ini dapat membantu dalam pemilihan algoritma lossless yang paling sesuai untuk digunakan dalam kompresi data.

## 2. Metode Penelitian

Penelitian dimulai dengan menentukan algoritma kompresi yang akan dibandingkan performanya. Kemudian memperisapkan file-file yang akan menjadi data untuk dikompresi. Tahap selanjutnya adalah pengujian performa dengan mengkompresi file-file menggunakan setiap algoritma yang sudah ditentukan, lalu melakukan perhitungan rasio kompresi dan penghematan ruang berdasarkan hasil pengujian. Pada bagian akhir, dilakukan pembahasan dan penarikan kesimpulan. Alur penelitian dapat dilihat pada Gambar 1.




***Gambar 1**. Alur Penelitian*

### 2.2 Algoritma kompresi

Algoritma kompresi yang akan diuji terhadap data adalah:

*2.1.1 LZMA2*

LZMA2 adalah algoritma kompresi data lossless. Ini adalah varian dari LZMA, yang dikembangkan oleh Igor Pavlov. LZMA adalah LZ77 yang ditingkatkan dan dioptimalkan. Prosedur kompresinya memiliki kesamaan antara algoritma deflate, namun menggunakan range encoding, sehingga encoder menjadi lebih kompleks untuk mencapai rasio kompresi yang lebih baik. Tingkat rasio kompresi LZMA yang tinggi ini pada dasarnya dihasilkan oleh dua konsep perhitungan, sliding dictionaries atau windows and Markov models. Fase pertama dari proses kompresi adalah pengkodean Lempel-Ziv yang menemukan dan mengurangi redundansi dengan mengubah potongan data menjadi pasangan distance-length. Kemudian fase kedua dilakukan, yang merupakan proses range encoding yang menggunakan model probabilitas khusus untuk data sampel yang berbeda. Algoritma LZMA2 adalah versi LZMA yang dimodifikasi dan ditingkatkan yang memberikan hasil multithreading yang lebih baik (Akoğuz et al., 2016).

*2.1.2 PPMd*

*Prediction by partial matching* (PPM) adalah algoritma kompresi data lossless. Ini adalah teknik kompresi data statistik adaptif berdasarkan pemodelan dan prediksi konteks. Prediksi dengan model pemetaan parsial menggunakan sekumpulan simbol sebelumnya dalam aliran simbol yang tidak terkompresi untuk memprediksi simbol berikutnya dalam aliran (K.Muthuchamy, 2018). PPMd merupakan implementasi dari PPMII oleh Dmitry Shkarin.

*2.1.3 BZip2*

*Burrows-Wheeler transform* (BWT) adalah metode kompresi data lossless yang pada dasarnya adalah algoritma pengurutan blok. Bagian transformasi tidak melakukan kompresi apa pun tetapi mengubah data input agar lebih mudah dikompresi. BZIP2 adalah kompresor data lossless open-source yang menggunakan algoritma Burrows-Wheeler dan pengkodean Huffman di latar belakang untuk kompresi (Akoğuz et al., 2016).

*2.1.4 Deflate*

Algoritma Deflate menggunakan pengkodean kamus LZ77 dan pengkodean entropi Huffman. Pengkodean LZ77 dan Huffman mengeksploitasi berbagai jenis redudansi untuk memungkinkan Deflate mencapai rasio kompresi yang tinggi. Namun, kebutuhan komputasi pengkode Huffman tinggi dan membuat kompresi Deflate agak lambat (Delaunay et al., 2019). Format data terkompresi DEFLATE terdiri dari serangkaian blok, sesuai dengan blok data masukan yang berurutan. Setiap blok dikompresi menggunakan kombinasi algoritma LZ77 dan pengkodean Huffman. Algoritme LZ77 menemukan substring berulang dan menggantinya dengan referensi mundur (offset jarak relatif). Algoritma LZ77 dapat menggunakan referensi ke string duplikat yang terjadi di blok yang sama atau sebelumnya (Oswal et al., 2016).

Data yang digunakan adalah file-file yang akan dikompresi oleh masing-masing algoritma. Variasi file yang akan digunakan ditentukan oleh (1) Tipe file: Tipe file yang akan dipilih adalah yang umum digunakan oleh pengguna biasa. Yaitu file teks, gambar, suara, dan video. (2) Ukuran file: File yang dipilih memiliki ukuran yang bervariasi, dari ukuran kecil hingga besar. Kriteria ukuran besar atau kecil disesuaikan dengan tipe filenya. Dan (3) Jumlah file: Algoritma kompresi akan diuji menggunakan lebih dari satu file.

File yang digunakan sebagai data, dipilih secara acak dengan tetap menyesuaikan terhadap tipe, ukuran dan jumlahnya. Pengujian performa dilakukan mengkompresi data menggunakan utilitas open source 7-Zip, yang menyediakan pilihan kompresi menggunakan berbagai algoritma. Perhitungan dilakukan terhadap hasil yang didapatkan.

### 2.3 Parameter performa algoritma

*2.3.1 Rasio kompresi*

Ini didefinisikan sebagai rasio dari jumlah total bit yang diperlukan untuk menyimpan data yang tidak terkompresi dan jumlah total bit yang diperlukan untuk menyimpan data terkompresi (Fitriya et al., 2017). Nilai rasio kompresi = 1 menunjukkan performa kompresi tidak baik, karena tidak ada perbedaan antara ukuran data yang belum dikompresi dan sudah dikompresi.

$$Rasio\ Kompresi = \frac{Jumlah\ bit\ data\ sebelum\ dikompresi}{Jumlah\ bit\ data\ setelah\ dikompresi}$$

*2.3.2 Penghematan ruang*

Ini didefinisikan sebagai pengurangan ukuran file relatif ke ukuran yang tidak terkompresi (Jayasankar et al., 2021). Nilai penghematan ruang = 0% menunjukkan performa kompresi tidak baik, karena tidak ada pengurangan ruang penyimpanan setelah data dikompresi.

$$Penghematan\ Ruang\ = 1 - \frac{Jumlah\ bit\ data\ setelah\ dikompresi}{Jumlah\ bit\ data\ sebelum\ dikompresi}$$

## 3. Hasil dan Pembahasan

Data yang digunakan untuk pengujian algoritma kompresi terbagi menjadi 4 tipe file, yaitu audio, gambar, dokumen, dan video. Keempat tipe file ini adalah tipe file yang biasa digunakan oleh pengguna komputer umum sehari-hari. Dari setiap tipe file, dipilih 3 format yang akan digunakan dalam pengujian performa. Pemilihan format ini juga mengacu kepada format yang sering ditemukan pengguna komputer pada umumnya. Berikut format-format dari masing-masing tipe file yang akan digunakan sebagai data dalam pengujian performa kompresi:

### 3.1 Audio

*3.1.1 Midi*

File MIDI (*Musical instrument Digital Interface*) adalah file digital yang berisi data musik. Ini bukan rekaman audio, melainkan serangkaian instruksi yang memberi tahu perangkat cara memutar musik.

*3.1.2 MP3*
File MP3 (MPEG audio Layer-3) adalah format file audio digital yang menggunakan algoritma kompresi lossy untuk mengurangi ukuran file

*3.1.3 WAV*
File WAV (*Waveform Audio File*), atau Wave, adalah format file audio digital yang tidak terkompresi, artinya menyimpan semua data audio asli tanpa kehilangan kualitas.

**3.2 Gambar**
*3.2.1 JPEG*
File JPG, atau JPEG *(Joint Photographic Experts Group)*, adalah format file gambar digital yang menggunakan algoritma kompresi lossy untuk mengurangi ukuran file.

*3.2.2PNG*
File PNG adalah file gambar yang disimpan dalam format *Portable Network Graphic* (PNG) menggunakan kompresi *lossless.*

*3.2. 3 TIFF*
File TIFF (Tag Image File Format) adalah format file gambar yang biasanya digunakan untuk menyimpan gambar berkualitas tinggi. File TIFF menggunakan kompresi lossless.

**3.3 Dokumen**
*3.3.1 TXT*
File TXT adalah file teks biasa yang hanya berisi teks dan tanpa pemformatan. Ini adalah jenis file teks paling dasar dan dapat dibuka dan diedit di editor teks apa pun.

*3.3.2 DOC/DOCX*
DOC adalah format dokumen yang digunakan oleh Microsoft Word, sedangkan DOCX adalah penggantinya. File DOC adalah file biner yang menyimpan teks, pemformatan, dan informasi lain tentang dokumen. File DOCX didasarkan pada format file Open XML, yang merupakan standar terbuka untuk menyimpan dokumen. File DOCX lebih kecil dan lebih efisien daripada file DOC, dan dapat dibuka dengan lebih banyak program pengolah kata.

*3.3.3 PDF*
File PDF (*Portable Document Format*) adalah file yang digunakan untuk mempresentasikan dokumen, termasuk pemformatan teks dan gambar, dengan cara yang tidak bergantung pada perangkat lunak aplikasi, perangkat keras, dan sistem operasi.

**3.4 Video**
*3.4.1 MP4*
MP4 (MPEG-4 Part 14) adalah format file yang umumnya digunakan untuk menyimpan video dan audio tetapi juga dapat digunakan untuk menyimpan subtitle dan gambar. Ini adalah format video yang sangat serbaguna dan terkompresi.

*3.4.2 AVI*
File AVI (*Audio Video Interleave*) adalah format file video yang dikembangkan oleh Microsoft pada tahun 1992. Ini adalah format wadah, yang artinya dapat menyimpan berbagai jenis data, termasuk video, audio, dan subtitle. File AVI biasanya digunakan untuk menyimpan video dan audio berkualitas tinggi, dan dapat diputar di berbagai perangkat.

*3.4.3 MKV*

MKV *(Matroska Multimedia Container*) adalah format wadah open source yang dapat menampung trek video, audio, gambar, atau subtitle dalam jumlah tak terbatas dalam satu file. File MKV biasanya digunakan untuk menyimpan video berkualitas tinggi.

Proses kompresi dilakukan terhadap setiap file menggunakan masing-masing algoritma kompresi. Ukuran file asli dan hasil kompresi masing-masing algoritma bisa dilihat pada Tabel 1.

*Tabel 1. Ukuran File Data (Bytes)*

| File | Data Asli | LZMA2 | PPMd | BZip2 | Deflate |
|---|---|---|---|---|---|
| *Audio1.mid* | 4.178 | 1.092 | 1.142 | 1.312 | 1.162 |
| *Audio2.mid* | 98.639 | 19.798 | 21.156 | 22.912 | 26.346 |
| *Audio3.mp3* | 5.139.375 | 4.986.398 | 5.009.149 | 4.977.303 | 5.007.598 |
| *Audio4.mp3* | 10.379.527 | 10.272.134 | 10.314.993 | 10.240.258 | 10.274.524 |
| *Audio5.wav* | 49.748.896 | 37.398.311 | 37.802.460 | 37.918.275 | 47.252.223 |
| *Audio6.wav* | 103.668.008 | 88.456.888 | 96.522.423 | 98.183.901 | 95.982.183 |
| *Image1.jpg* | 143.683 | 141.560 | 143.466 | 142.503 | 141.296 |
| *Image2.jpg* | 4.258.964 | 4.250.189 | 4.235.714 | 4.242.021 | 4.245.538 |
| *Image3.png* | 12.021.324 | 12.021.019 | 12.041.055 | 12.022.691 | 12.025.930 |
| *Image4.png* | 24.488.979 | 24.453.100 | 24.570.460 | 24.494.845 | 24.478.416 |
| *Image5.tif* | 32.734.500 | 32.031.888 | 32.117.947 | 32.304.986 | 32.305.358 |
| *Image6.tif* | 55.823.800 | 54.937.663 | 55.327.542 | 55.539.092 | 54.890.239 |
| *Doc1.txt* | 3.951 | 1.914 | 1.663 | 1.938 | 1.766 |
| *Doc2.txt* | 101.886 | 30.286 | 25.159 | 28.244 | 32.349 |
| *Doc3.docx* | 6.117.697 | 5.826.924 | 5.917.816 | 5.891.727 | 5.862.214 |
| *Doc4.doc* | 11.023.360 | 9.447.233 | 9.689.765 | 9.753.495 | 9.611.665 |
| *Doc5.pdf* | 27.423.931 | 22.042.292 | 22.205.666 | 22.730.103 | 22.603.262 |
| *Doc6.pdf* | 102.680.854 | 101.684.239 | 103.270.139 | 102.573.392 | 102.002.109 |
| *Video1.mp4* | 5.799.572 | 5.592.781 | 5.644.034 | 5.568.403 | 5.579.972 |
| *Video2.mp4* | 15.789.173 | 15.589.753 | 15.817.179 | 15.580.809 | 15.585.047 |
| *Video3.avi* | 58.818.752 | 58.503.097 | 59.191.527 | 58.847.862 | 58.707.739 |
| *Video4.avi* | 106.017.048 | 102.355.667 | 100.338.737 | 102.009.944 | 104.453.446 |
| *Video5.mkv* | 546.394.849 | 546.330.641 | 557.439.729 | 548.109.264 | 546.483.677 |
| *Video6.mkv* | 1.050.521.600 | 1.050.419.210 | 1.071.664.085 | 1.053.764.651 | 1.050.759.768 |
| Gabungan | 2.229.200.825 | 2.186.829.382 | 2.229.376.291 | 2.204.998.455 | 2.207.983.067 |

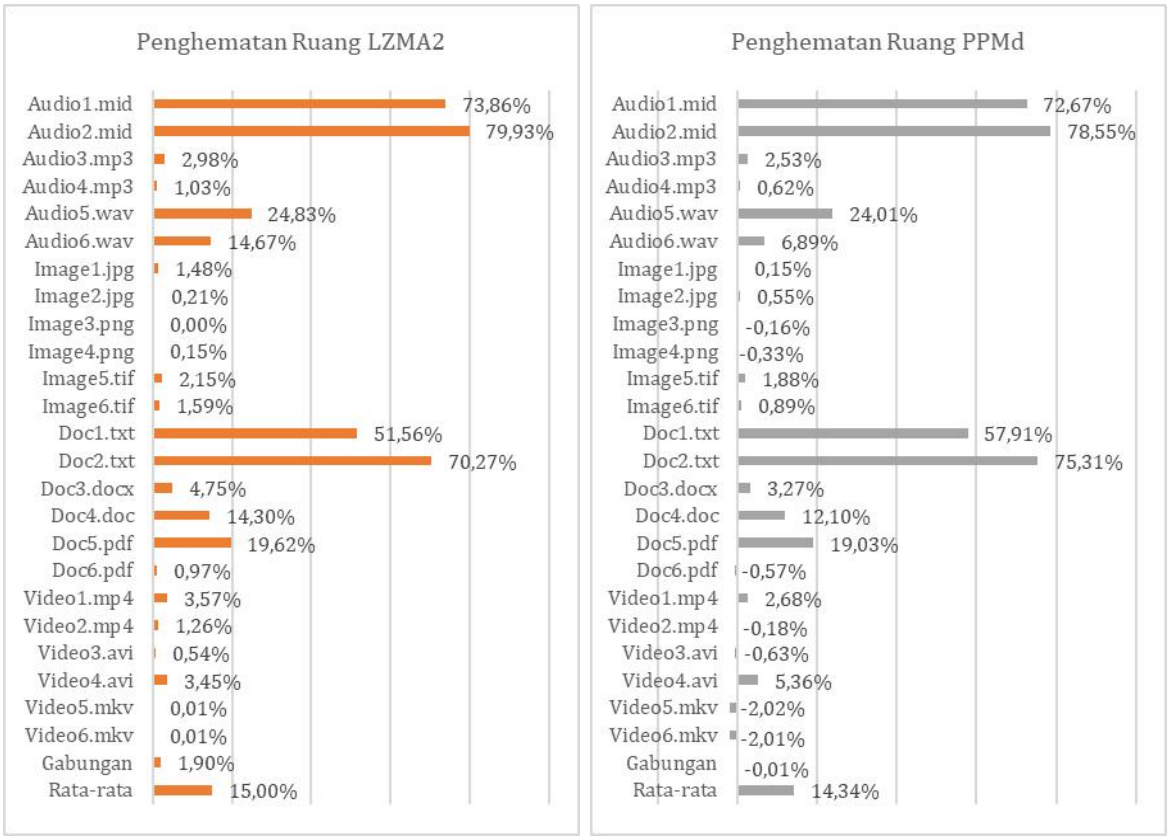
Ukuran file ini kemudian digunakan untuk menghitung rasio kompresi dan penghematan ruang. Nilai rasio kompresi setiap file, bisa dilihat pada Tabel 2.

*Tabel 2. Rasio Kompresi*

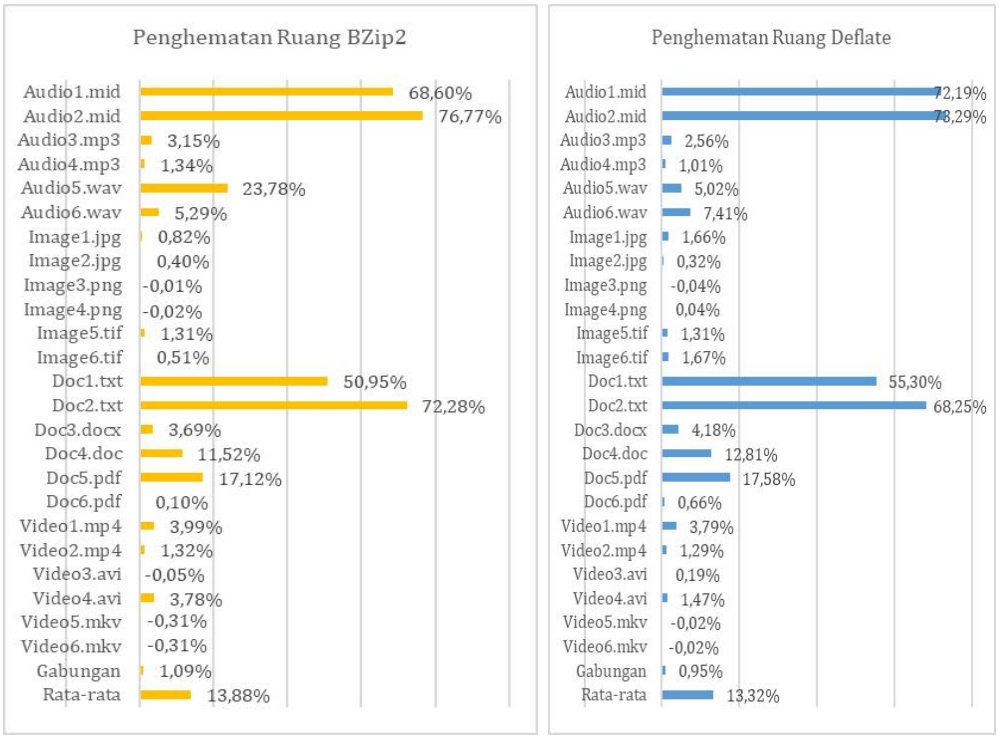
| File | Data Asli | LZMA2 | PPMd | BZip2 | Deflate |
|---|---|---|---|---|---|
| Audio1.mid | 1,000 | 3,826 | 3,658 | 3,184 | 3,596 |
| Audio2.mid | 1,000 | 4,982 | 4,662 | 4,305 | 3,744 |
| Audio3.mp3 | 1,000 | 1,031 | 1,026 | 1,033 | 1,026 |

| | | | | | |
|---|---|---|---|---|---|
| Audio4.mp3 | 1,000 | 1,010 | 1,006 | 1,014 | 1,010 |
| Audio5.wav | 1,000 | 1,330 | 1,316 | 1,312 | 1,053 |
| Audio6.wav | 1,000 | 1,172 | 1,074 | 1,056 | 1,080 |
| Image1.jpg | 1,000 | 1,015 | 1,002 | 1,008 | 1,017 |
| Image2.jpg | 1,000 | 1,002 | 1,005 | 1,004 | 1,003 |
| Image3.png | 1,000 | 1,000 | 0,998 | 1,000 | 1,000 |
| Image4.png | 1,000 | 1,001 | 0,997 | 1,000 | 1,000 |
| Image5.tif | 1,000 | 1,022 | 1,019 | 1,013 | 1,013 |
| Image6.tif | 1,000 | 1,016 | 1,009 | 1,005 | 1,017 |
| Doc1.txt | 1,000 | 2,064 | 2,376 | 2,039 | 2,237 |
| Doc2.txt | 1,000 | 3,364 | 4,050 | 3,607 | 3,150 |
| Doc3.docx | 1,000 | 1,050 | 1,034 | 1,038 | 1,044 |
| Doc4.doc | 1,000 | 1,167 | 1,138 | 1,130 | 1,147 |
| Doc5.pdf | 1,000 | 1,244 | 1,235 | 1,207 | 1,213 |
| Doc6.pdf | 1,000 | 1,010 | 0,994 | 1,001 | 1,007 |
| Video1.mp4 | 1,000 | 1,037 | 1,028 | 1,042 | 1,039 |
| Video2.mp4 | 1,000 | 1,013 | 0,998 | 1,013 | 1,013 |
| Video3.avi | 1,000 | 1,005 | 0,994 | 1,000 | 1,002 |
| Video4.avi | 1,000 | 1,036 | 1,057 | 1,039 | 1,015 |
| Video5.mkv | 1,000 | 1,000 | 0,980 | 0,997 | 1,000 |
| Video6.mkv | 1,000 | 1,000 | 0,980 | 0,997 | 1,000 |
| Gabungan | 1,000 | 1,019 | 1,000 | 1,011 | 1,010 |
| **Rata-rata** | **1,000** | **1,457** | **1,465** | **1,402** | **1,377** |

Rata-rata rasio kompresi tertinggi dihasilkan oleh algoritma PPMd yaitu sebesar 1,465, dan terendah oleh Deflate sebesar 1,377. Meskipun begitu, nilai rata-rata rasio kompresi semua algoritma tidak terpaut jauh, dengan range = 1,465 – 1.377 = 0,088. Ditemukan bahwa untuk format file tertentu, seperti midi (*audio*) dan *text* (dokumen), nilai rasio kompresi jauh lebih baik dibandingkan rata-rata. Ini dapat disebabkan oleh format file tersebut yang tidak memiliki konten yang kompleks dan memiliki banyak redudansi. Untuk file-file lainnya yang memiliki rasio kompresi rendah bahkan hingga kurang dari 1, dapat disebabkan karena banyak format file yang pada dasarnya sudah terkompresi, sehingga penerapan kompresi lebih lanjut tidak bisa memberikan hasil yang baik.




***Gambar 2**. Penghematan Ruang LZMA2 dan PPMd*




***Gambar 3**. Penghematan Ruang BZip2 dan Deflate*

Nilai penghematan ruang, yang dapat dilihat pada Gambar 2 dan Gambar 3, menunjukkan bahwa rata-rata penghematan ruang terbaik dihasilkan oleh algoritma LZMA2 yaitu sebesar 15,00%, dan terburuk oleh Deflate sebesar 13,32%. Serupa dengan rasio kompresi, nilai rata-rata penghematan ruang semua algoritma tidak terpaut jauh dengan range = 15,00% - 13.32% = 1,68%, dan format file midi (audio) dan text (dokumen), mendapatkan nilai penghematan ruang yang jauh lebih baik dibandingkan rata-rata. Terdapat juga penghematan ruang dengan nilai negatif, yang mengartikan bahwa ukuran file setelah dikompresi malah menjadi lebih besar dari sebelum dikompresi.

*Tabel 3. Perbandingan Nilai Rata-rata Rasio Kompresi dan Penghematan Ruang Terhadap Nilai Tertinggi*

| | LZMA2 | PPMd | BZip2 | Deflate |
|---|---|---|---|---|
| **Rasio Kompresi** | 0,994 | 1,000 | 0,957 | 0,940 |
| **Pengehematan Ruang** | 1,000 | 0,956 | 0,925 | 0,887 |
| **Rata-rata** | **0,997** | **0,978** | **0,941** | **0,914** |

Tabel 3 menunjukkan perbandingan nilai rata-rata dari rasio kompresi terhadap nilai tertingginya, yaitu yang dihasilkan algoritma PPMd, dan perbandingan nilai rata-rata penghematan ruang terhadap nilai tertingginya yang dihasilkan algoritma LZMA2. Perbandingan ini digunakan untuk menentukan algoritma yang terbaik dari kombinasi rasio kompresi dan penghematan ruang.

Pada dasarnya, baik pada rasio kompresi maupun penghematan ruang, angka yang lebih tinggi menunjukkan performa kompresi yang lebih baik. Dari hasil yang ditemukan, dapat diberikan analisis sebagai berikut. Performa hasil kompresi lebih ditentukan oleh format file, dibanding tipe file. Ini karena format file berbeda memiliki karakteristik yang berbeda, yang membuatnya lebih atau kurang dapat menerima kompresi. Misalnya, file teks umumnya dapat dikompresi dengan baik karena berisi banyak pola yang berulang (Sayood, 2012). Rata-rata nilai performa kompresi tidak terlalu jauh. Dengan melihat nilai rasio kompresi dan penghematan ruang, secara umum LZMA2 adalah yang terbaik, diikuti oleh PPMd, BZip2, dan yang terendah adalah Deflate. Meskipun nilai rata-rata rasio kompresi PPMd lebih baik dari LZMA2, tapi perbandingannya lebih kecil dari perbandingan nilai rata-rata penghematan ruang, di mana LZMA2 lebih baik dari PPMd.

Khusus untuk tipe file text (dokumen), PPMd mendapatkan hasil terbaik dibanding yang lain. Ini adalah hasil yang tidak mengejutkan, karena PPMd diketahui sebagai algoritma kompresi data open-source yang sangat efektif untuk file teks, dan merupakan versi yang lebih baik dari algoritma PPM. Dan hasil penelitian menunjukkan PPM sebagai algoritma kompresi yang sangat baik digunakan untuk file teks (Hutagalung, 2018). Ada beberapa file yang ukuran hasil kompresinya malah lebih besar dari ukuran data aslinya. Penjelasan untuk ini adalah karena banyak format file yang pada dasarnya sudah terkompresi. Dan jika operasi kompresi selalu dapat membuat ukuran file menjadi lebih kecil, maka jika dilakukan berulang dan terus menerus, pada akhirnya kompresi dapat membuat sebuah file berukuran 0 byte dan tetap menyimpan semua data aslinya. Ini adalah hal yang tidak mungkin. Data yang sudah dikompresi umumnya merupakan kandidat yang buruk untuk kompresi lebih lanjut.

## 4. Kesimpulan

Perbandingan performa kompresi data terhadap algoritma kompresi lossless LZMA2, PPMd, BZip2, Deflate, menggunakan nilai Rasio Kompresi dan Penghematan Ruang, menunjukkan bahwa LZMA2 mencapai performa kompresi yang lebih baik dibanding yang lain. Meskipun begitu, ada beberapa format file yang bisa dikompresi lebih baik menggunakan algoritma selain LZMA2. Karena itu, penting untuk diingat bahwa hasil pengujian ini baiknya hanya digunakan sebagai gambaran umum dalam memilih algoritma kompresi. Rasio kompresi dan penghematan ruang yang bisa dicapai secara spesifik akan bergantung pada data yang digunakan. Pada akhirnya, cara terbaik menentukan algoritma kompresi yang paling tepat digunakan untuk mengkompresi sebuah atau sekumpulan data, adalah dengan mencobanya dan memilih hasil yang terbaik.

Untuk penelitian lanjutan, ada baiknya parameter pengujian dan jumlah data ditambah. Penambahan parameter seperti kecepatan waktu kompresi, dapat membuat keputusan pemilihan algoritma kompresi lebih bernuansa, karena akan ada pertimbangan mengenai keseimbangan antara pengurangan ukuran data dan waktu yang dibutuhkan untuk mencapainya. Penambahan jumlah data dapat memberikan akurasi dan keyakinan yang lebih baik terhadap hasil penelitian karena mengurangi kemungkinan hasil yang disebabkan oleh kebetulan. Mengikutsertakan algoritma-algoritma kompresi data lain yang belum digunakan pada penelitian ini juga bisa dilakukan agar komparasi algoritma menjadi lebih komprehensif

## 5. Referensi


Akoğuz, A., Bozkurt, S., Gözütok, A., Alp, G., Turan, E., Bogaz, M., & Kent, S. (2016). Comparison Of Open Source Compression Algorithms On Vhr Remote Sensing Images For Efficient Storage HIERARCHY. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3–9. https://doi.org/http://dx.doi.org/10.5194/isprsarchives-XLI-B4-3-2016

Delaunay, X., Courtois, A., & Gouillon, F. (2019). Evaluation of lossless and lossy algorithms for the compression of scientific datasets in netCDF-4 or HDF5 files. *Geoscientific Model Development (GMD)*, *12*(9), 4099–4113. https://doi.org/https://doi.org/10.5194/gmd-12-4099-2019

Fitriya, L. A., Purboyo, T. W., & Prasasti, A. L. (2017). A Review of Data Compression Techniques. *International Journal of Applied Engineering Research*, *12*(19), 8956–8963.

Gupta, A., Bansal, A., & Khanduja, V. (2017). Modern lossless compression techniques: Review, comparison and analysis. *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*. https://doi.org/https://doi.org/10.1109/ICECCT.2017.8117850

Haque, M. J., & Huda, M. N. (2017). Study on Data Compression Technique. *International Journal of Computer Applications*, *159*(5), 6–13. https://doi.org/http://dx.doi.org/10.5120/ijca2017912416

Hosseini, M. (2012). A Survey of Data Compression Algorithms and their Applications. *Applications of Advanced Algorithm*. https://doi.org/http://dx.doi.org/10.13140/2.1.4360.9924

Hutagalung, R. (2018). Implementasi Algoritma Prediction By Partial Matching (Ppm) Pada Kompresi File Teks Terenkripsi Elgamal. *JURIKOM (Jurnal Riset Komputer)*, *5*(6), 611–620. https://doi.org/http://dx.doi.org/10.30865/jurikom.v5i6.1196

Jayasankar, U., Thirumal, V., & Ponnurangam, D. (2021). A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications. *Journal of King Saud University - Computer and Information Sciences*, *33*(2), 119–140. https://doi.org/https://doi.org/10.1016/j.jksuci.2018.05.006

K.Muthuchamy. (2018). A Study on Various Data Compression Types and Techniques. *International Journal of Research and Analytical Reviews (IJRAR)*, *5*(3), 945–950.

Oswal, S., Singh, A., & Kumari, K. (2016). Deflate Compression Algorithm. *International Journal of Engineering Research and General Science*, *4*(1), 430–436.

Sayood, K. (2012). Dictionary Techniques. In *Introduction to Data Compression* (4th ed., pp. 135–136). Elsevier.

Taylor, P. (2022). *Total data volume worldwide 2010-2025 | Statista*. Statista. https://www.statista.com/statistics/871513/worldwide-data-created/

Usama, M., Malluhi, Q. M., Zakaria, N., Razzak, I., & Iqbal, W. (2021). An efficient secure data compression technique based on chaos and adaptive Huffman coding. *Peer-to-Peer Networking and Applications*, *14*(3), 2651–2664. https://doi.org/https://doi.org/10.1007/s12083-020-00981-8

Yang, Y., Mandt, S., & Theis, L. (2023). An Introduction to Neural Data Compression. *Foundations and Trends in*