

Analisis Perbandingan Algoritma Taboo Code Dan Shannon Fano Pada Kompresi File Audio

Andika Anastasya Sitorus

Teknik Informatika, Universitas Budi Darma, Indonesia

e-mail: ta552436@gmail.com

Keywords:

*Compression,
Audio File,
Taboo Code,
Shannon-Fano,
Comparison.*

ABSTRACT

Audio file compression plays a critical role in digital data management, particularly in improving storage efficiency and transmission performance. However, selecting an appropriate compression algorithm remains challenging due to the diverse characteristics of audio signals, including frequency complexity and acceptable levels of data loss. This study aims to analyze and compare the performance of two compression algorithms, namely Taboo Code and Shannon-Fano, in compressing various types of audio files. An experimental approach was employed by testing both algorithms on different audio datasets, including music, speech, and sound effects. The evaluation parameters include compression ratio, data loss level, and processing time. The results indicate that Taboo Code achieves higher compression ratios under certain conditions, making it more efficient in terms of storage. In contrast, Shannon-Fano demonstrates better performance in terms of compression speed and lower data loss in several scenarios. This study provides practical insights for selecting suitable compression algorithms based on specific application requirements, whether prioritizing storage efficiency, processing speed, or output quality.

Kata Kunci:

*Kompresi,
File Audio,
Taboo Code,
Shanon Fano,
Perbandingan.*

ABSTRAK

Kompresi file audio merupakan aspek penting dalam pengelolaan data digital, khususnya untuk meningkatkan efisiensi penyimpanan dan transmisi. Namun, pemilihan algoritma kompresi yang tepat menjadi tantangan karena karakteristik sinyal audio yang beragam, seperti kompleksitas frekuensi dan toleransi terhadap kehilangan data. Penelitian ini bertujuan untuk membandingkan kinerja dua algoritma kompresi, yaitu Taboo Code dan Shannon-Fano, dalam mengompresi berbagai jenis file audio. Metode yang digunakan adalah pendekatan eksperimental dengan menguji kedua algoritma pada beberapa jenis audio, termasuk musik, suara percakapan, dan efek suara. Parameter evaluasi meliputi rasio kompresi, tingkat kehilangan data, dan waktu kompresi. Hasil penelitian menunjukkan bahwa Taboo Code memiliki keunggulan dalam menghasilkan rasio kompresi yang lebih tinggi pada kondisi tertentu. Sementara itu, Shannon-Fano menunjukkan performa yang lebih baik dalam hal kecepatan kompresi dan tingkat kehilangan data yang lebih rendah. Penelitian ini memberikan kontribusi dalam membantu pemilihan algoritma kompresi yang optimal sesuai dengan kebutuhan aplikasi, baik yang berorientasi pada efisiensi penyimpanan maupun kualitas hasil kompresi.

Korespondensi Penulis *):

Andika Anastasya Sitorus
Universitas Budi Darma

Jalan Sisingamangaraja No 338, Kecamatan Medan Kota, Kota Medan, Indonesia.

Diajukan: 27-11-2025 | Direvisi: 10-12-2025 | Diterima: 18-12-2025 | Diterbitkan: 29-12-2025

1. PENDAHULUAN

Perkembangan teknologi digital telah mendorong peningkatan kebutuhan terhadap pengelolaan data yang efisien, khususnya pada data multimedia seperti audio. File audio, terutama dalam format MP3, umumnya memiliki ukuran yang relatif besar sehingga memerlukan ruang penyimpanan yang signifikan serta waktu transmisi yang lebih lama. Kondisi ini menuntut adanya teknik kompresi data yang mampu mengurangi ukuran file tanpa menghilangkan informasi penting di dalamnya.

Kompresi data merupakan proses pengkodean ulang data menjadi representasi yang lebih ringkas dengan tujuan utama mengurangi redundansi. Teknik kompresi dapat diterapkan pada berbagai jenis data, termasuk teks, gambar, dan audio. Dalam konteks audio, pendekatan kompresi lossless menjadi penting karena memungkinkan data dikembalikan secara utuh tanpa kehilangan informasi. Berbagai algoritma kompresi telah dikembangkan dengan karakteristik dan performa yang berbeda-beda, sehingga pemilihan algoritma yang tepat menjadi aspek krusial dalam implementasinya.

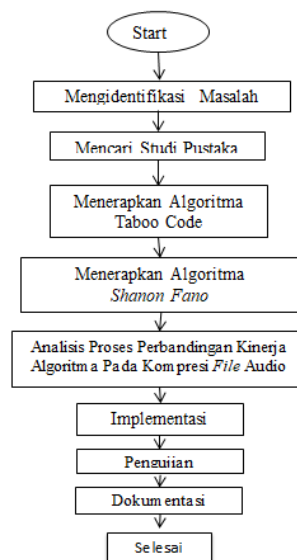
Penelitian ini berfokus pada analisis perbandingan dua algoritma kompresi berbasis entropy coding, yaitu Taboo Code dan Shannon-Fano. Algoritma Taboo Code menggunakan pendekatan panjang kode variabel berdasarkan pola bilangan tertentu, sedangkan Shannon-Fano membangun kode berdasarkan probabilitas kemunculan simbol, di mana simbol dengan frekuensi tinggi direpresentasikan dengan kode yang lebih pendek. Kedua algoritma ini memiliki keunggulan dan keterbatasan masing-masing dalam hal rasio kompresi, efisiensi waktu, dan kualitas hasil kompresi.

Beberapa penelitian terdahulu menunjukkan bahwa algoritma kompresi memiliki performa yang bervariasi tergantung pada jenis data yang digunakan. Namun demikian, masih terdapat kebutuhan untuk melakukan analisis komparatif secara spesifik pada file audio guna memperoleh pemahaman yang lebih komprehensif terkait efektivitas masing-masing algoritma. Selain itu, penggunaan metode eksponensial sebagai pendekatan pengambilan keputusan juga menjadi nilai tambah dalam menentukan algoritma yang paling optimal berdasarkan kriteria tertentu. Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk menganalisis dan membandingkan performa algoritma Taboo Code dan Shannon-Fano dalam kompresi file audio berbasis lossless. Parameter yang digunakan meliputi rasio kompresi, efisiensi ruang penyimpanan, dan waktu proses. Hasil penelitian diharapkan dapat memberikan kontribusi dalam menentukan algoritma kompresi yang paling sesuai untuk kebutuhan aplikasi audio digital serta menjadi referensi bagi pengembangan sistem kompresi di masa mendatang.

2. METODE PENELITIAN

2.1 Tahapan Penelitian

Sebelum membuat kerangka penelitian, peneliti harus melakukan analisis. Tahapan ini mempermudah proses penelitian dan menghasilkan hasil yang optimal. Tahapan penelitian ini digambarkan dalam gambar 1 sebagai berikut:



Gambar 1. Tahapan Penelitian

Berikut adalah penjelasan dari tahapan-tahapan penelitian:

1. Mengidentifikasi masalah
Langkah pertama adalah memaparkan sumber permasalahan mengenai kapasitas penyimpanan yang penuh serta lambatnya pengiriman file audio.
2. Mencari studi pustaka
Penelitian kepustakaan adalah proses membaca berbagai sumber referensi, seperti buku, majalah, internet, dan lainnya, untuk memahami subjek penelitian.
3. Analisis Penerapan Algoritma *Taboo Code* dan *Shanon Fano*
Algoritma *taboo code* dan *Shanon Fano* untuk langkah ini menerapkan kompresi *lossless*. Dimana data yang sudah dikompres bisa dikompres kembali.
4. Perbandingan Kinerja Algoritma Pada Kompresi *file* Audio
Proses perbandingan ini dilakukan untuk mengetahui bagaimana proses kompresi *file* audio dengan algoritma *taboo code* dan *shannon fano* bekerja dengan tujuan mengurangi ukuran data.

5. Implementasi

Selama pengembangan dan pengkodean perangkat lunak, langkah yang akan diambil adalah implementasi, yang akan menentukan apakah sistem berfungsi dengan baik.

6. Pengujian

Pada tahapan ini, pengujian sistem dilakukan dengan menggunakan Algoritma *Taboo Code* dan *Shannon Fano*.

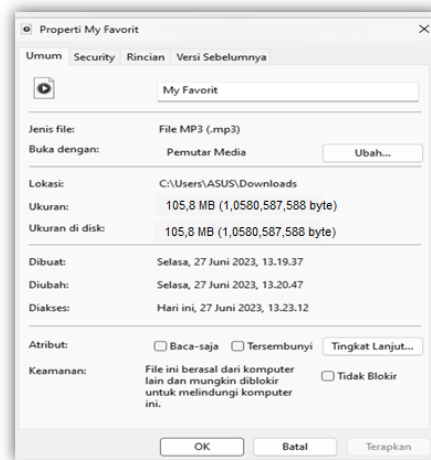
7. Dokumentasi

Langkah terakhir dalam melakukan penelitian dalam bentuk laporan adalah dokumentasi. Langkah ini menjelaskan desain yang digunakan, yang dimaksudkan untuk memudahkan pembaca dalam mengembangkan aplikasi tambahan.

Tabel 1. File Audio yang kan Dikompresi

| | |
|-------------|------------|
| Name file | My Favorit |
| Jenis file | MP3 |
| Ukuran file | 105,8 MB |

Berikut file gambar yang digunakan sebagai sampel data dengan *properties* yang dapat dilihat pada gambar 2 berikut



Gambar 2. properties file

Saat mengompresi data, beberapa algoritma dapat digunakan sebagai metode untuk melakukan studi kasus. Berbagai faktor digunakan untuk menentukan kualitas kompresi, yaitu [1]:

1. *Ratio Of Compression* (RC)

Ratio Of Compression (RC) adalah nilai yang membandingkan ukuran bit data sebelum kompresi dengan ukuran bit data sebelum kompresi dan ukuran bit data yang dikompresi. Berikut diuraikan:

$$RC = \frac{\text{Ukuran data sebelum di kompresi}}{\text{Ukuran data setelah di kompresi}} \dots\dots\dots(1)$$

2. *Compression Ratio* (CR)

Compression RATIO(CR) adalah perbandinganpersentaseantara data terkompresi dan tidak terkompresi. Berikut diuraikan

$$CR = \frac{\text{Ukuran data setelah kompresi}}{\text{Ukuran setelah dikompresi}} \times 100\% \dots\dots\dots(2)$$

3. *Redudancy* (RD)

Redudancy adalah hasil penelitian nilai rasio komparasi, Secara sistematis dapatdituliskan sebagai berikut:

$$RD = \frac{\text{File sebelum di kompresi} - \text{File setelah di kompresi}}{\text{Ukuran file sebelum di kompres}} \times 100\% \dots\dots\dots(3)$$

4. *Space Saving* (Ss)

Space Saving (Ss) adalah selisih antara data yang belum dikompresi denganbesar data yang sudah dikompresi.

$$Ss = 100\% - CR \dots\dots\dots(4)$$

File audio adalah informasi multimedia serta lokasi penyimpanan *file* audio di sistem komputer. *File* audio atau *file* audio dengan beberapa format seperti MP3, WAV, AIFF, dll. *File* audio juga dapat menerapkan teknik kompresi untuk mengurangi ukuran *file* ke ukuran yang lebih kecil. [9]. *File* audio juga merupakan salah satu *file* yang biasa digunakan dalam transmisi data. *File* audio seringkali berukuran sangat besar, sehingga sulit untuk mengirimkan dan menyimpan data. [10].

2.2. Algoritma Taboo Code

Algoritma *taboo code* berfokus pada panjang variabel. Prinsipnya adalah untuk memiliki bilangan n positif dan membalikkan pola n untuk menunjukkan bahwa kode telah berakhir [1].

1. Jenis algoritma *taboo* yang pertama berbasis blok dan memiliki panjang kelipatan n . Algoritma *taboo* integer berbasis blok adalah urutan blok n -bit di mana pengguna memilih nilai n dan blok terakhir memiliki pola tabu yang tidak dapat muncul di blok lainnya. Sebuah blok yang terdiri dari n bit memiliki nilai $2n$, dan jika satu nilai dicadangkan, maka setiap blok kode yang tersisa dapat memiliki salah satu blok kode tersebut dalam pola $2n - 1$ bit.
2. Algoritma *taboo* jenis kedua tidak memiliki batasan kelipatan untuk panjang kode secara keseluruhan. Hubungannya dengan bilangan *fibonacci* ke- n menunjukkan bahwa tipe ini tidak terikat.

2.3. Algoritma Shanon Fano

Algoritma *shanon fano*, variable *I-word* digunakan, dan beberapa simbol dalam data diwakili dengan kode simbol yang lebih pendek dari jumlah simbol yang ada dalam data. Kemungkinan kode simbol yang lebih pendek terkait dengan kemungkinan bahwa simbol tersebut ada dalam data. [11].

Algoritma *shannon fano* menghasilkan kode simbol dengan panjang yang berbeda, yang membuatnya unik dan dapat dikodekan [12]. Cara efisien lainnya dalam variable *length-coding* dan *shanon fano encoding*. Prosedur dalam *shanon fano encoding* adalah:

1. Mengurutkan probabilitas simbol dari sumber tertinggi ke sumber terendah.
2. Membaginya menjadi dua bagian yang sama besar, dengan bagian atas diberi nilai 0 dan bagian bawah diberi nilai 1.
3. Ulangi langkah kedua dengan membagi setiap pembagian dengan probabilitas yang sama, probabilitas yang sama, probabilitas yang sama, dan tidak mungkin lagi.
4. *encoding* setiap simbol asli sumber menjadi urutan biner yang dihasilkan oleh setiap proses pembagian tersebut.

3. HASIL DAN ANALISIS

3.1 Hasil

Hasil kompresi string bit yang lebih sedikit dari 96 string bit awal dari *file* audio yang dikompresi menggunakan algoritma *taboo codes* ditunjukkan di bawah ini:

01001100100001000101001000011000100001000111000100000011.

3.2 Analisis

Proses yang terlibat dalam sistem dan pengoperasian kompresi *file* audio dalam penelitian ini. Pada gambar di bawah, contoh *file* audio dikompresi dengan membandingkan kode tabu dan algoritma Shanon-Fano dan menghasilkan output terkompresi dalam karakter/symbol tunggal. *shanon-fano* untuk mengembalikan *file* asli dalam bentuk tertulis.



Gambar 3. General Architecture System

1. Penerapan Algoritma Taboo Codes

Sebelum melakukan kompresi *file*, Anda harus membaca biner *file* audio untuk mendapatkan data berupa data biner. Anda dapat menggunakan aplikasi Binary Viewer untuk membaca biner *file* audio dan mengetahui nilai binernya.

Tabel 2. String karakter yang belum dikompresi

| M | Karakter | Frekuensi | ASCII (binary) | Bit | Frek×Bit |
|---|----------|-----------|----------------|-----|----------|
| 1 | 01 | 4 | 00000001 | 8 | 32 |
| 2 | BD | 1 | 10111101 | 8 | 8 |
| 3 | 00 | 8 | 00000000 | 8 | 64 |
| 4 | 6D | 1 | 01101101 | 8 | 8 |

| | | | | | |
|-----------|----|---|----------|-----|---|
| 5 | 43 | 1 | 10101100 | 8 | 8 |
| 6 | 55 | 1 | 01010101 | 8 | 8 |
| Total Bit | | | | 128 | |

Satu karakter memiliki nilai biner 8 bit, sehingga string dengan dua belas karakter memiliki nilai biner 128 bit, menurut tabel 2. Algoritma kode tabu terlebih dahulu mengurutkan jumlah karakter yang muncul di setiap kompresi. Adapun *string* bit sebelum dikompresi adalah sebagai berikut: 00000001101111010000000011011010110001101010101.

Setelah menentukan kemunculan setiap karakter dan menemukan nilai binernya, langkah berikutnya adalah mengurutkan setiap karakter berdasarkan frekuensi teringginya. Setelah mengurutkan setiap karakter berdasarkan frekuensi teringginya, nilai biner sebelumnya diganti dengan kode kebenaran algoritma kode tabu, yang diurutkan berdasarkan kemunculannya.

Tabel 3. Data Setelah Dikompresi Menggunakan Algoritma *Taboo Codes*

| N | Char | Biner | Bit | Frek | Bit × Frek |
|-----------|------|--------|-----|------|------------|
| 1 | 01 | 0100 | 4 | 4 | 16 |
| 2 | 00 | 1000 | 4 | 8 | 32 |
| 3 | BD | 1100 | 4 | 1 | 4 |
| 4 | 6D | 010100 | 6 | 1 | 6 |
| 5 | 43 | 011000 | 6 | 1 | 6 |
| 6 | 55 | 011100 | 6 | 1 | 6 |
| Total Bit | | | | | 70 |

Setelah *diencode* menggunakan code kebenaran algoritma *taboo codes*, proses kompresi harus disusun kembali sesuai dengan posisi karakter pada string, seperti yang ditunjukkan dalam tabel 3.

Tabel 4. *String bit* hasil kompresi dengan algoritma *Taboo Codes*

| | | | | |
|------|--------|------|------|--------|
| 01 | BD | 00 | 01 | 6D |
| 0100 | 1100 | 0000 | 0100 | 010100 |
| 01 | 43 | 00 | 01 | 55 |
| 1000 | 101011 | 0000 | 0100 | 011100 |

Setelah setiap karakter disusun kembali menggunakan kode kebenaran algoritma *taboo codes*, langkah berikutnya adalah menyusun kembali hasil kompresi setiap karakter, yang menghasilkan string bit berikut: 0100 1100 1000 0100 010100 1000 101011 1000 0100 011100.

Aturan 8-bit digunakan untuk menyesuaikan data untuk penyimpanan bit-string hingga kelipatan 8, Kemudian untuk mengetahui jumlah padding dan penandaan yang akan ditambahkan selama proses jumlah semua *string* bit yang dikompresi akan dibagi 8. String bit 00000001 akan dijumlahkan jika sisa penjumlahan bit string yang lebih besar dari 8 adalah 0, dan padding dan penandaan akan ditambahkan jika panjang bit string melebihi 8 yang tidak dapat dibagi.

Tabel 5. Penambahan *padding* dan *flagging*

| Padding | Flagging |
|--------------------|-------------|
| $7 - n + "1"$ | $9 - n$ |
| $7 - 6 + "1" = 01$ | $9 - 6 = 3$ |
| | 00000011 |

Dengan demikian, string bit berikut diperoleh: 0100 1100 1000 0100 010100 1000 011000 1000 0100 01110001 00000011, dengan total bit yang *dipadding* dan *diflagging*. *String* bit awal panjangnya adalah 54 setelah *dipadding* dan *diflagging*, dan string bit total yang dikelompokkan menjadi 8 bit. Selanjutnya, *string* bit yang dikelompokkan diubah menjadi bilangan desimal, bilangan desimal yang dihasilkan dari perubahan ini adalah bilangan desimal yang dihasilkan dari perubahan ini.

Tabel 6. Pengelompokan Bit

| No | String bit hasil kompresi | Bilangan Desimal |
|----|---------------------------|------------------|
| 1 | 01001100 | 76 |
| 2 | 10000100 | 132 |
| 3 | 01010010 | 82 |
| 4 | 00011000 | 24 |
| 5 | 10000100 | 130 |

| | | |
|---|----------|-----|
| 6 | 01110001 | 113 |
| 7 | 00000011 | 3 |

Hasil kompresi string bit yang lebih sedikit dari 96 string bit awal dari *file* audio yang dikompresi menggunakan algoritma *taboo codes* ditunjukkan di bawah ini:

01001100100001000101001000011000100001000111000100000011.

2. Penerapan Algoritma Shanon-Fano

Target implementasi adalah untuk memberikan gambaran umum tentang apa yang akan dilakukan. Hasil analisis akan digunakan sebagai dasar untuk membuat program yang memenuhi persyaratan sistem. Contoh mengkodekan *file audio* "01BD006D4355". Kodekan setiap karakter dalam ASCII.

Tabel 7. kode char dalam ASCII

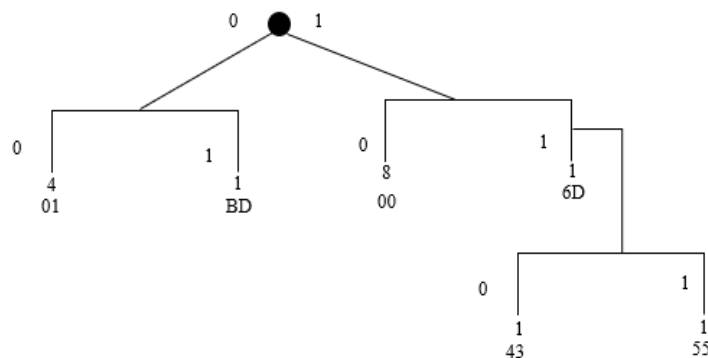
| Karakter | Binner | Frekuensi |
|----------|----------|-----------|
| 01 | 00000001 | 4 |
| BD | 10111101 | 1 |
| 00 | 00000000 | 8 |
| 6D | 01101101 | 1 |
| 43 | 10101100 | 1 |
| 55 | 01010101 | 1 |

Jika *file audio* hexsadecimal 1 diubah dalam bit, menjadi : 00000001 10111101

00000000 01101101 01100011 01010101= total **48 bit**.

1. Pembentukan Pohon Binner

Untuk membuat pohon biner, tabel karakter yang diurutkan dibagi menjadi dua bagian, dengan frekuensi masing-masing subbagian mendekati atau sama. Subpohon kiri diberi label 0 dan subpohon kanan diberi label 1. Ulangi pembagian ini secara rekursif hingga pengkodean yang sesuai diterapkan pada setiap karakter. Proses pengulangan pembagian pohon disajikan pada lampiran.



Gambar 4. Pohon Shannon-Fano yang telah Terbentuk

2. Pembentukan Kode Shannon Fano

Kode *shannon fano* ditunjukkan oleh barisan angka 0 dan 1 dari sisi pohon dari akar ke daun. Untuk membentuk kode *shannon fano*, pohon biner diikuti dari akar ke daun.

Tabel 8. Hasil Pengkodean

| Karakter | Kode Shannon Fano |
|----------|-------------------|
| 01 | 00 |
| BD | 01 |
| 00 | 10 |
| 6D | 11 |

| | |
|----|-----|
| 43 | 110 |
| 55 | 111 |

Tabel di bawah ini menunjukkan hasil perhitungan setelah kompresi.

Tabel 9. Perhitungan

| Karakter | Frekuensi | Kode Shannon Fano | Panjang Kode Shannon Fano | Total Bit (Panjang Kode x Frekuensi) |
|--------------|-----------|-------------------|---------------------------|--------------------------------------|
| 01 | 4 | 00 | 2 | 6 |
| BD | 1 | 01 | 2 | 2 |
| 00 | 8 | 10 | 2 | 16 |
| 6D | 1 | 11 | 2 | 2 |
| 43 | 1 | 110 | 3 | 3 |
| 55 | 1 | 111 | 3 | 3 |
| Total | 13 | | | 31 |

Tabel di atas menunjukkan kode setiap simbol dalam pesan, yang menghasilkan jumlah bit yang diperlukan untuk mewakili pesan tersebut, yaitu 31 bit. Dengan menggunakan ASCII 8 bit, diperlukan $13 \times 8 \text{ bit} = 104 \text{ bit}$, sehingga rasio kompresi adalah $31/104 \times 100\% = 29,81\%$. Dari perhitungan ini, dapat disimpulkan bahwa metode *shannon fano* dapat dikompresi menjadi 29,81% pada file audio "01BD006D4355".

3. Menentukan hasil keputusan

Setelah melakukan perhitungan, langkah selanjutnya yaitu melakukan perhitungan nilai prioritas keputusan dari metode algoritma *taboo code* dan algoritma *shannon fano*.

Tabel 4.11 Prioritas Keputusan

| No | Alternatif | Total Nilai | Rangking |
|----|-------------------------------|-------------|----------|
| 1. | Algoritma <i>Taboo Code</i> | 45,313% | 1 |
| 2. | Algoritma <i>Shannon Fano</i> | 29,81% | 2 |

Berdasarkan Analisa di atas disimpulkan bahwa algoritma *taboo code* lebih efektif dan optimal dalam mengkompresi file audio dibandingkan dengan algoritma *shannon fano*.

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa algoritma *Taboo Code* dan *Shannon-Fano* berhasil diimplementasikan dalam proses kompresi file audio berformat MP3. Proses pengujian dan perbandingan kedua algoritma dilakukan dengan mengacu pada beberapa parameter evaluasi, yaitu rasio kompresi, efisiensi ukuran file, dan waktu pemrosesan, yang dihitung berdasarkan formula yang telah ditetapkan dalam sistem. Hasil analisis menunjukkan bahwa aplikasi yang dirancang menggunakan Visual Studio 2010 mampu menjalankan proses kompresi dan perbandingan algoritma secara optimal. Dari hasil perbandingan tersebut, algoritma *Taboo Code* terbukti memiliki kinerja yang lebih unggul dibandingkan *Shannon-Fano*, khususnya dalam menghasilkan kompresi yang lebih efektif dan optimal pada file audio. Dengan demikian, algoritma *Taboo Code* dapat direkomendasikan sebagai metode yang lebih sesuai untuk digunakan dalam kompresi file audio berbasis lossless pada penelitian ini.

REFERENSI

- [1] S. Simanjuntak, "Implementasi Metode *Taboo Code* Untuk Kompresi File Video," vol. 2, no. 1, pp. 32–38, 2022.
- [2] N. Yuni and I. Hutagalung, "Penerapan Algoritma *Taboo Codes* Pada Kompresi File Text," vol. 1, no. 2, pp. 43–49, 2022.
- [3] I. A. Febriansya, "Implementasi Kompresi Audio Menggunakan Algoritma *Shannon-Fano* Pada Aplikasi Podcast," *Setrum Sist. Kendali-Tenaga- elektronika-telekomunikasi-komputer*, vol. 10, no. 1, pp. 35–48, 2021, doi: 10.36055/setrum.v10i1.11171.
- [4] S. H. Silitonga and S. D. Nasution, "Implementasi Algoritma *Boldi-Vigna Codes* Untuk Kompresi File Audio Pada Aplikasi Pemutar Audio Berbasis Web," vol. 6, no. November, 2022, doi: 10.30865/komik.v6i1.5754.
- [5] N. Aisyah and S. Aripin, "Penerapan Algoritma *Elias Omega Code* Pada Kompresi File Audio Aplikasi Murottal Muzzamil Hasbalah," vol. 9, pp. 113–119, 2020.

- [6] A. M. Ruslida, Sapri, and D. Sartika, "Implementasi Algoritma Byte Pair Encoding Untuk Kompresi File," vol. 18, no. 2, pp. 253–260, 2022.
- [7] R. Yanur Tanjung, "Perancangan Aplikasi Kompresi File Dokumen Menggunakan Algoritma Additive Code," *J. Ris. Komputer*, vol. 8, no. 4, pp. 2407–389, 2021, doi: 10.30865/jurikom.v8i4.3593.
- [8] L. Y. Telaumbanua, E. Bu, and K. Ulfa, "Implementasi Algoritma Code- Excited Linear Prediction (CELP) pada Kompresi File Audio," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 6, no. November, pp. 317–321, 2022, doi: 10.30865/komik.v6i1.5701.
- [9] L. Lahmuddin, "Penerapan Algoritma Boldy Vigna Untuk Mengkompresi Pada File Audio Aplikasi Kajian Dan Murottal Islami," vol. 1, no. 1, pp. 1–9.
- [10] R. O. Finola, "Penerapan Algoritma Interpolative Coding Untuk Kompresi File Audio," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 3, no. 1, pp. 378–384, 2019, doi: 10.30865/komik.v3i1.1616.
- [11] K. Widatama and W. T. Saputro, "Perbandingan Kinerja Algoritma Huffman dan Algoritma Shannon-Fano Dalam Mengkompresi File Citra," vol. 2, no. November, 2019.
- [12] M. Apriyanto and Hutrianto, "Analisa Penerapan Algoritma Goldbach Codes dan Metode Shannon-Fano Pada Kompresi File Teks," *Bina Darma Conf. Comput. Sci.*, vol. 2, no. 5, pp. 207–218, 2020, [Online]. Available: <https://conference.binadarma.ac.id/index.php/BDCCS/article/download/1714/771>
- [13] A. Rhamadani, "Analisa Perbandingan Algoritma Taboo Codes Dan Algoritma Yamamoto ' s Recursive Code Untuk Kompresi File Teks Menggunakan Metode Exponential," vol. 6, no. November, pp. 140–150, 2022, doi: 10.30865/komik.v6i1.5728.
- [14] B. A. Silaban, "Analisa Kompresi File Teks Dengan Kombinasi Metode Burrows-Wheeler Transform Dan Shannon-Fano," vol. 6, no. November, pp. 707–715, 2022, doi: 10.30865/komik.v6i1.5760.
- [15] O. Lawolo, R. K. Hondro, and K. Ulfa, "Analisa Perbandingan Implementasi Algoritma Set Partitioning in Hierarchical Trees (SPIHT) dan Boldy Vigna Code untuk Kompresi File Audio," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 6, no. November, pp. 431–435, 2022, doi: 10.30865/komik.v6i1.5694.
- [16] I. Lestari, "Analisa Perbandingan Algoritma Goldbach Codes Dengan Algoritma Sequitur Pada Kompresi File Text Menggunakan Metode ...," *Pelita Inform. Inf. dan ...*, vol. 8, pp. 15–18, 2019, [Online]. Available: <https://www.ejurnal.stmik-budidarma.ac.id/index.php/pelita/article/view/1520>.