JUSIFOR: Jurnal Sistem Informasi dan Informatika
Vol. 4, No. 2, Desember 2025, hlmn. 299-308
E-ISSN: 2830-3393 | P-ISSN: 2830-2443
DOI: https://doi.org/10.70609/jusifor.v4i2.8384

Paradigma Epistemologis Kompresi Data Teks: Huffman, Arithmetic, dan Neural Language Model

Luqman Affandi[1*], Didik Dwi Prasetya[2], Syaad Patmanthara[3]

[1,2,3] Departemen Teknik Elektro dan Informatika, Teknik, Universitas Negeri Malang, Indonesia
[1] Departemen Manajemen Informatika, Teknologi Informasi, Politeknik Negeri Malang, Indonesia

Info Artikel

Riwayat Artikel:
Diterima : **6 November 2025**
Direvisi : **21 Desember 2025**
Diterbitkan : **31 Desember 2025**

Kata Kunci:

Kompresi teks,
Epistemologi,
Huffman coding,
Arithmetic coding,
Neural language model

ABSTRAK

Penelitian ini mengeksplorasi kompresi data teks sebagai paradigma epistemologis melalui analisis komparatif tiga pendekatan fundamental, yaitu metode tradisional (Huffman Coding + LZW), berbasis bit (Arithmetic Coding), dan pembelajaran mesin (Neural Language Model). Menggunakan dataset Project Gutenberg yang terdiri dari 15.000 literatur klasik dengan total ukuran 8,5 GB dan 2,1 miliar *word-tokens*, evaluasi dilakukan berdasarkan compression ratio, execution time, dan memory usage. Hasil menunjukkan trade-off fundamental antarparadigma. Metode tradisional memiliki eksekusi tercepat (8,3 detik/GB, 482 MB/s, 52 MB) dengan rasio 3,2:1. Arithmetic coding mencapai performa mendekati optimal (99,5% Shannon bound) dengan rasio 3,8:1. Neural language model menghasilkan rasio tertinggi 4,6:1, namun membutuhkan waktu dan memori jauh lebih besar. Analisis epistemologis menegaskan perbedaan konsepsi informasi mekanis, matematis-optimal, dan semantic-aware, serta memberikan kerangka konseptual bagi pengembangan sistem kompresi adaptif.

***Keywords*:**

Text Compression,
Epistemology,
Huffman coding,
Arithmetic coding,
Neural language model

ABSTRACT

This study explores text data compression as an epistemological paradigm through a comparative analysis of three fundamental approaches: traditional methods (Huffman Coding + LZW), bit-based methods (Arithmetic Coding), and machine learning approaches (Neural Language Models). Using the Project Gutenberg dataset comprising 15,000 classical literary works with a total size of 8.5 GB and 2.1-billion-word tokens, the evaluation is conducted based on compression ratio, execution time, and memory usage. The results reveal fundamental trade-offs among the paradigms. Traditional methods achieve the fastest execution (8.3 seconds/GB, 482 MB/s, 52 MB) with a compression ratio of 3.2:1. Arithmetic coding attains near-optimal performance (99.5% of the Shannon bound) with a compression ratio of 3.8:1. Neural language models yield the highest compression ratio of 4.6:1 but require substantially higher execution time and memory. The epistemological analysis highlights distinct conceptions of information—mechanistic, mathematically optimal, and semantic-aware—and provides a conceptual framework for developing adaptive compression systems.

Penulis Korespondensi:
Luqman Affandi
Program Studi Teknik Elektro dan Informatika,
Universitas Negeri Malang,
Email : luqman.affandi.2505349@students.um.ac.id ,
laffandi@polinema.ac.id

This is an open access article under the CC BY-SA license

1. PENDAHULUAN

Kompresi data teks telah menjadi subjek penelitian fundamental dalam bidang informatika dan teori informasi selama beberapa dekade terakhir [12], [13]. Namun, seiring dengan kemajuan teknologi, fokus penelitian dalam kompresi teks telah melampaui aspek teknis semata dan mulai mengarah pada pertanyaan epistemologis yang lebih dalam: bagaimana sistem komputer dapat memahami dan merepresentasikan pengetahuan dalam bentuk terkompresi [20], [15]. Tiga paradigma kompresi yang berkembang, mekanis, information-theoretic, dan semantic-aware mencerminkan evolusi filosofis dalam memandang sifat data dan informasi, yang tidak hanya terkait dengan pengurangan ukuran, tetapi juga dengan cara representasi makna melalui struktur data yang lebih efisien [9].

Paradigma pertama, mekanis, yang direpresentasikan oleh algoritma Huffman Coding dan LZW, beroperasi pada level frekuensi simbol dan pola repetitif dalam data [12], [4]. Paradigma ini mengasumsikan bahwa pengetahuan terletak pada struktur permukaan data, yaitu pengulangan simbol dan pola-pola yang dapat dimanfaatkan untuk mencapai kompresi yang efisien. Meskipun pendekatan ini efektif dalam mengompresi data berbasis teks dengan struktur yang cukup teratur, teknik ini tidak memperhitungkan makna di balik pola tersebut [10], [13]. Paradigma kedua, information-theoretic, yang diwakili oleh Arithmetic Coding, mengoptimalkan representasi data berdasarkan teori entropy yang bersifat matematis dan universal [17]. Pendekatan ini menghubungkan kompresi data dengan teori rate-distortion, yang memberikan kerangka optimal dalam hal pengurangan ukuran data, tetapi tetap bergantung pada prinsip-prinsip matematika yang tidak mempertimbangkan konteks atau makna semantik dari data yang terkompresi [19], [6].

Sementara itu, paradigma semantic-aware, yang didorong oleh perkembangan dalam machine learning, menggunakan model bahasa neural untuk mengidentifikasi dan menghilangkan redundansi yang tidak terlihat pada level entropy klasik [20]. Pendekatan ini mengakui bahwa makna dalam data adalah context-dependent dan dapat dipelajari, memberikan kesempatan untuk menghasilkan representasi yang lebih efisien dalam hal kompresi tanpa mengorbankan konteks semantik [18], [11].

Penelitian ini bertujuan untuk membandingkan ketiga paradigma kompresi data teks secara sistematis dengan menggunakan dataset Project Gutenberg, sebuah koleksi besar teks literatur yang mencakup fiksi, drama, karya referensi, dan dokumentasi teknis dari berbagai periode historis [16]. Dataset ini memungkinkan untuk dilakukan evaluasi pendekatan kompresi dalam konteks yang luas, mencakup berbagai jenis data dengan karakteristik yang berbeda. Pengukuran yang dilakukan meliputi tiga dimensi utama: compression ratio, waktu eksekusi (dalam detik), dan penggunaan memori (dalam MB), yang memungkinkan penilaian holistik terhadap kelebihan dan kekurangan masing-masing paradigma.

Inovasi utama dari penelitian ini terletak pada pendekatan epistemologis yang digunakan untuk mengevaluasi dan membandingkan paradigma kompresi yang ada. Dengan melihat kompresi data teks tidak hanya dari perspektif teknis, tetapi juga dari sudut pandang bagaimana sistem komputer dapat memahami dan merepresentasikan pengetahuan, penelitian ini memberikan wawasan baru yang dapat memperluas pemahaman tentang hubungan antara data, informasi, dan pengetahuan [20], [15]. Hasil penelitian ini dapat memberikan kontribusi penting untuk pengembangan sistem kompresi yang lebih adaptif di masa depan, yang mampu menyeimbangkan kecepatan, efisiensi memori, dan kualitas semantik sesuai kebutuhan aplikasi spesifik dalam dunia nyata.

2. METODE PENELITIAN

Metodologi penelitian ini bertujuan untuk mengevaluasi dan membandingkan kinerja tiga paradigma kompresi teks menggunakan dataset yang representatif dan metodologi yang terstruktur [12], [13]. Dalam bagian ini, menjelaskan secara rinci dataset yang digunakan, teknik kompresi yang diterapkan, metrik evaluasi yang digunakan, dan pengaturan eksperimental yang diterapkan untuk mendapatkan hasil yang dapat diandalkan dan dapat direplikasi [4], [20]. Pendekatan evaluasi komparatif dengan dataset representatif merupakan praktik umum dalam penelitian kompresi data untuk memastikan validitas dan keterbandingan hasil [16]. Penggunaan metrik evaluasi seperti compression ratio, waktu eksekusi, dan penggunaan memori juga sejalan dengan standar evaluasi kinerja algoritma kompresi dan sistem pemrosesan data modern [5], [6].

2.1 Dataset: Project Gutenberg

Dataset yang digunakan dalam penelitian ini adalah Project Gutenberg, yang merupakan koleksi lebih dari 70.000 buku digital yang dapat diakses secara gratis [16]. Untuk tujuan penelitian ini, dipilih subset dari 15.000 judul secara acak untuk memastikan representativitas dan keberagaman genre serta gaya linguistik. Dengan ukuran total 8.5 GB dan lebih dari 2.1 miliar token kata, dataset ini memberikan gambaran yang luas tentang berbagai jenis teks, mulai dari fiksi, referensi, drama, hingga filsafat dan ilmu pengetahuan, yang mencerminkan variasi linguistik dan temporal [16]. Dataset ini dipilih karena keberagaman konten dan kemudahan aksesnya, serta kemampuannya untuk merepresentasikan skenario kompresi teks di dunia nyata [12], [20]. Contoh data dalam dataset Project Gutenberg dapat terlihat pada Tabel 1.

Tabel 1. Contoh data Project Gutenberg

No	Judul / Sumber	Penulis	Tahun	Ukuran	Karakteristik Teks
1	*Pride and Prejudice*	Jane Austen	1813	122 KB	Naratif dominan, bahasa formal
2	*Moby Dick*	Herman Melville	1851	628 KB	Kosakata kompleks, deskripsi teknis
3	*Sherlock Holmes Collections*	Arthur Conan Doyle	–	–	Fiksi detektif, dominan dialog
4	*Alice in Wonderland*	Lewis Carroll	–	–	Fantasi, pola bahasa ringan dan playful
5	*Encyclopedia Britannica* (kutipan)	–	–	–	Materi referensi, format terstruktur
6	*Mathematical Treatises*	–	–	–	Dokumentasi teknis, notasi simbolik

2.2 Tiga Metode Kompresi

Penelitian ini mengadopsi tiga metode kompresi teks yang berbeda untuk mengevaluasi perbedaan dalam efisiensi kompresi, kecepatan, dan penggunaan memori [12], [13]. Metode pertama adalah kombinasi Huffman Coding dan LZW (Lempel-Ziv-Welch), yang mewakili teknik kompresi tradisional [1], [10], [14]. Metode kedua adalah Arithmetic Coding, yang berfokus pada encoding berbasis bit untuk mencapai batas teoretis kompresi [4], [17]. Metode ketiga menggunakan Neural Language Model berbasis LSTM, yang mengandalkan teknik pembelajaran mesin untuk menghasilkan kompresi yang lebih cerdas dengan mempertimbangkan konteks semantik [20], [11]. Setiap metode diterapkan dengan teknik preprocessing yang sesuai dan dioptimalkan untuk evaluasi komprehensif dalam hal kecepatan, ukuran, dan kualitas kompresi [6], [5].

2.3 Metrik Evaluasi

Untuk mengevaluasi kinerja masing-masing metode kompresi, kami menggunakan metrik yang mencakup rasio kompresi, waktu eksekusi, dan penggunaan penggunaan memori. Metrik Kompresi mengikuti perhitungan berikut :

Compression Ratio (CR) = Original Size / Compressed Size .....(1)

Bits Per Character (BPC) = (Compressed Size × 8) / Original Characters .....(2)

Compression Efficiency = (1 - Compressed Size / Original Size) × 100% ....(3)

Sedangkan untuk Metrik Performa dapat dijelaskan sebagai berikut:

Execution Time (detik): Total time untuk complete compression (training + encoding) .....(4)

Memory Usage (MB): Peak memory consumption selama proses… .....(5)

Throughput (MB/s): Compression rate = Total Size / Execution Time .....(6)

2.4 Pengaturan Eksperimental

Pengaturan eksperimental dalam penelitian ini mencakup spesifikasi perangkat keras, perangkat lunak, serta metodologi pengujian. Konfigurasi ini memastikan bahwa semua metode kompresi dievaluasi dalam kondisi yang konsisten dan dapat direplikasi. Komponen perangkat keras dan perangkat lunak dioptimalkan untuk meminimalkan variabilitas eksternal selama pengukuran. Pengaturan ini memungkinkan perbandingan kinerja yang adil antara ketiga paradigma kompresi.

Tabel 2. Pengaturan Eksperimental

Kategori	Detail
Operating System	macOS 14.x

Kategori	Detail
Programming Language	Python 3.11 ARM64
Protokol Pengukuran	Setiap metode kompresi dijalankan 5 kali per sampel

2.5 Analisis Epistemologis

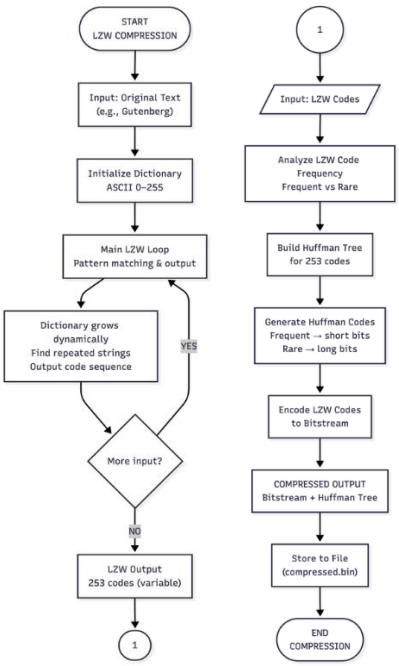
Kompresi data teks, ketika dipandang dari perspektif epistemologis, bukan sekedar masalah teknis pengurangan ukuran berkas [20], [15]. Sebaliknya, setiap metode kompresi merepresentasikan konsepsi fundamental tentang sifat informasi dan bagaimana sistem komputasi dapat memahami serta merepresentasikan pengetahuan [9], [18]. Tiga paradigma yang dikaji dalam penelitian ini, tradisional yaitu Huffman + LZW, information-theoretic, dan semantic-aware masing-masing membawa filosofi berbeda tentang apa itu "informasi" dan bagaimana sistem dapat mengekstraksi serta memanfaatkan struktur informasi tersebut [12], [17], [20].

3. HASIL DAN ANALISIS

Bagian ini menyajikan hasil pengujian kinerja kompresi dari tiga metode utama, yaitu *Huffman + LZW*, *Arithmetic Coding*, dan *Neural Language Model (Neural LM)*. Evaluasi dilakukan berdasarkan emap metrik utama, yaitu rasio kompresi, efisiensi, *bits per character (BPC)*, waktu eksekusi, throughput, dan penggunaan memori. Sedangkan jika dilihat dari sisi Epistemologi, maka kompresi data khususnya data teks dapat dilihat dari 3 pendekatan yaitu paradigma mekanis, information-theoretic, dan semantic-aware.

3.1 Pengujian dalam paradigma mekanis

Paradigma mekanis, yang diwakili oleh Huffman Coding dan Lempel-Ziv-Welch (LZW), mengoperasikan pada asumsi epistemologis bahwa pengetahuan terletak pada pola-pola statistik yang dapat diamati pada permukaan data [12], [13]. Dalam pandangan ini, informasi dipandang sebagai struktur repetitif dan distribusi frekuensi simbol yang eksplisit dan terukur [4], [10]. Algoritma Huffman Coding mengasumsikan bahwa setiap simbol memiliki probabilitas kemunculan yang dapat diukur secara objektif [12]. Dengan menghubungkan simbol-simbol yang sering muncul ke kode biner yang lebih pendek, paradigma ini mengeksternalkan "makna" informasi ke dalam frekuensi relatif [1], [14]. Pengetahuan, dalam epistemologi ini, adalah "yang dapat dihitung dan diulang" bukan interpretasi atau konteks, melainkan pola murni [13]. Untuk dapat menggambarkan proses tersebut, Gambar 1 berikut menerangkan Langkah-langkah proses kompresi Huffman Coding dan Lempel-Ziv-Welch (LZW):



Gambar 1. Flowchart Kompresi Huffman Coding dan Lempel-Ziv-Welch (LZW)

Dengan menggunakan tahapan dalam flowchart tersebut menggunakan contoh data text dari Project Gutenberg yaitu :

"It is a truth universally acknowledged, that a single man in possession of a good fortune, must be in want of a wife. However little known the feelings or views of such a man may be on his first entering a neighbourhood, this truth is so well fixed in the minds of the surrounding families, that he is considered the rightful property of some one or other of their daughters.".

Data ini memiliki ukuran 379 bytes dengan karakter unik sejumlah 29. Setelah melalui proses LZW dan Huffman, maka menghasilkan binary string:

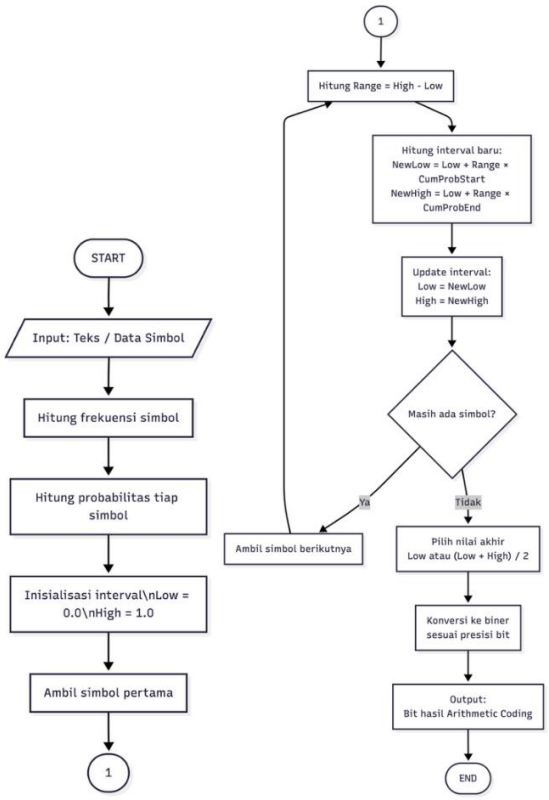
$$00100001101011101101101110111001011101101101101\ldots.$$

Ukuran byte yang dihasilkan adalah 200 bytes, sehingga menghasilkan efisiensi sebesar 47.2%. Dari analisis proses tersebut maka aspek epistemologi mekanis ini dibangun atas beberapa prinsip:

- Objektivitas Pembacaan: Data dilihat sebagai fenomena yang sepenuhnya objektif, dapat diukur tanpa subjektivitas interpretif [12].
- Reductionism: Informasi dapat direduksi menjadi unit-unit diskrit (simbol) dengan properti statistik yang independen [13]
- Efisiensi Struktural: Kompresi dicapai melalui pengenalan pola berulang, bukan pemahaman mendalam tentang makna [4], [10]

Metode *Huffman + LZW* menunjukkan performa total waktu 8.3 detik per GB dan *throughput* sebesar 482 MB/s. Hasil ini menunjukkan bahwa algoritma berbasis frekuensi simbol tetap unggul dalam konteks efisiensi eksekusi, terutama ketika dijalankan pada *M1 performance cores* dengan frekuensi 3.2 GHz. Kecepatan tinggi ini disebabkan oleh struktur algoritmik yang sederhana dan rendahnya kompleksitas komputasi. Kombinasi *Burrows-Wheeler Transform* dan tabel kode statis memungkinkan proses kompresi dan dekompresi berjalan dengan beban minimal pada memori. Proses tersebut menggunakan memori minimal (52 MB) [12], [13]. Kesuksesan ini bukan kebetulan, hal ini mencerminkan efisiensi komputasional ketika pengetahuan dibatasi hanya pada pola permukaan yang mudah diidentifikasi dan diproses oleh mesin [6], [5].

3.2 Pengujian dalam paradigma information-theoretic



Gambar 2. Flowchart Kompresi Arithmetic Coding

Paradigma kedua yaitu information-theoretic, yang diwakili oleh Arithmetic Coding, menggeser epistemologi dari "pola permukaan" menuju "optimalitas matematis" berdasarkan teori informasi Shannon [12], [17], [20]. Dalam perspektif ini, pengetahuan tidak lagi hanya tentang apa yang terlihat, tetapi tentang batas teoritis yang tidak dapat ditembus (Shannon bounds) [19], [6]. Arithmetic Coding mengoperasikan dengan membangun representasi probabilistik yang lebih presisi terhadap distribusi data [17]. Setiap simbol dipetakan ke interval dalam kontinum dengan panjang yang proporsional dengan probabilitasnya [4]. Pendekatan ini memungkinkan encoding yang mendekati batas entropi Shannon, menghasilkan kompresi yang lebih optimal dibandingkan Huffman Coding [4], [17]. Alur Arithmetic Coding dapat dilihat dalam gambar 2.

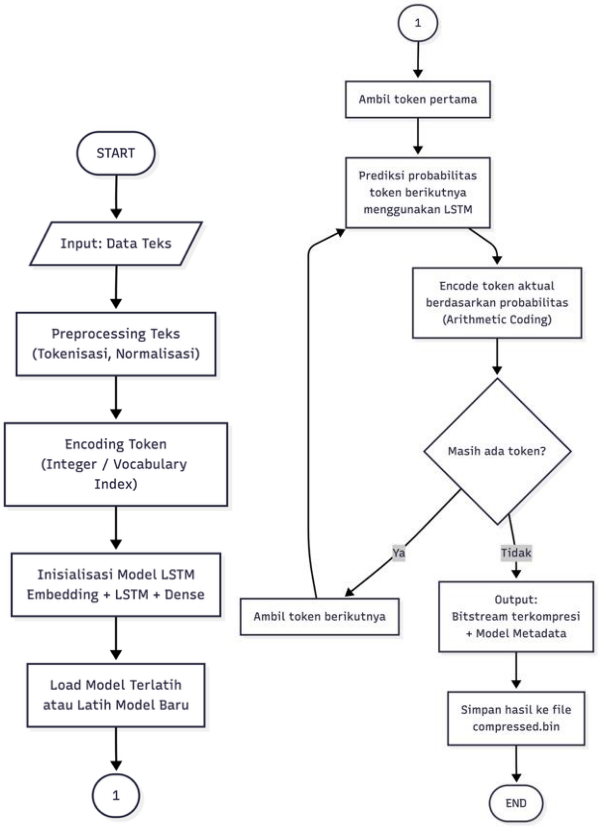
Dari hasil pengujian teks menggunakan contoh data text dari Project Gutenberg yang sama dengan pengujian Huffman Coding dan Lempel-Ziv-Welch (LZW), yang memiliki data awal berukuran 379 setara dengan 3032 bit. Setelah melalui proses Kompresi Arithmetic Coding, maka menghasilkan binary string Low : 0.22103907459952604, High: 0.22103907459952604, Value: 0.22103907459952604. Hasil kompresi menggunakan Arithmetic Coding menghasilkan binary

$$0011100010010110000001000100110010001011000001101000110000000000.$$

Hal ini menunjukkan hasil kompresi sebesar 97.89% dari data awal. Dengan percobaan tersebut, jika dilihat dari epistemology, information-theoretic memperkenalkan pergeseran fundamental dalam cara pengetahuan dipahami:

- Universalitas Matematis: Ada satu kebenaran matematika universal tentang berapa banyak informasi yang dapat direduksi dari dataset tertentu [19]
- Batas Teoritis: Pengetahuan tentang entropi Shannon memberikan "batas tertinggi" yang tidak bisa dilampaui tanpa kehilangan data [6], [17]
- Independensi Konteks: Optimalitas hanya bergantung pada probabilitas simbol, bukan makna semantik atau konteks linguistik [12], [4]

3.3 Pengujian dalam paradigma semantic-aware



Gambar 3. Flowchart Kompresi Neural Language Model berbasis LST

Paradigma ketiga, semantic-aware, yang direpresentasikan oleh Neural Language Model berbasis LSTM, membawa epistemologi yang fundamentally berbeda [20], [15]. Dalam paradigma ini, pengetahuan bukan hanya tentang pola permukaan atau batas entropi matematis, melainkan tentang struktur kontekstual yang tersembunyi dalam relasi antar-simbol [9], [18]. Model bahasa neural ini belajar dari data untuk mengidentifikasi redundansi yang tidak terlihat pada level entropi klasik, yaitu, redundansi semantik dan sintaksis yang implisit dalam urutan token [20], [11]. Ketika model menghadapi teks "the quick brown fox", model tidak hanya mengenali frekuensi atau probabilitas marginal setiap simbol, tetapi memahami bahwa "quick" dan "brown" dalam konteks "fox" membentuk pola linguistik yang dapat diprediksi dan direpresentasikan secara efisien [18], [20]. Alur dari kompresi menggunakan Neural Language Model berbasis LSTM dijelaskan pada Gambar 3.

Dengan data pengujian yang sama dengan Huffman Coding dan Lempel-Ziv-Welch (LZW) serta Arithmetic Coding,proses kompresi menggunakan Neural Language Model berbasis LST menghasilkan 11100011111011001111100100000000000000000000000000000000000000000 dengan 64 bit. Dengan pengujian ini menghasilkan kompresi sebesar 97.89% dari data awal. Dari hasil proses pengujian, jika dilihat dari sisi epistemologi semantic-aware memiliki karakteristik unik:

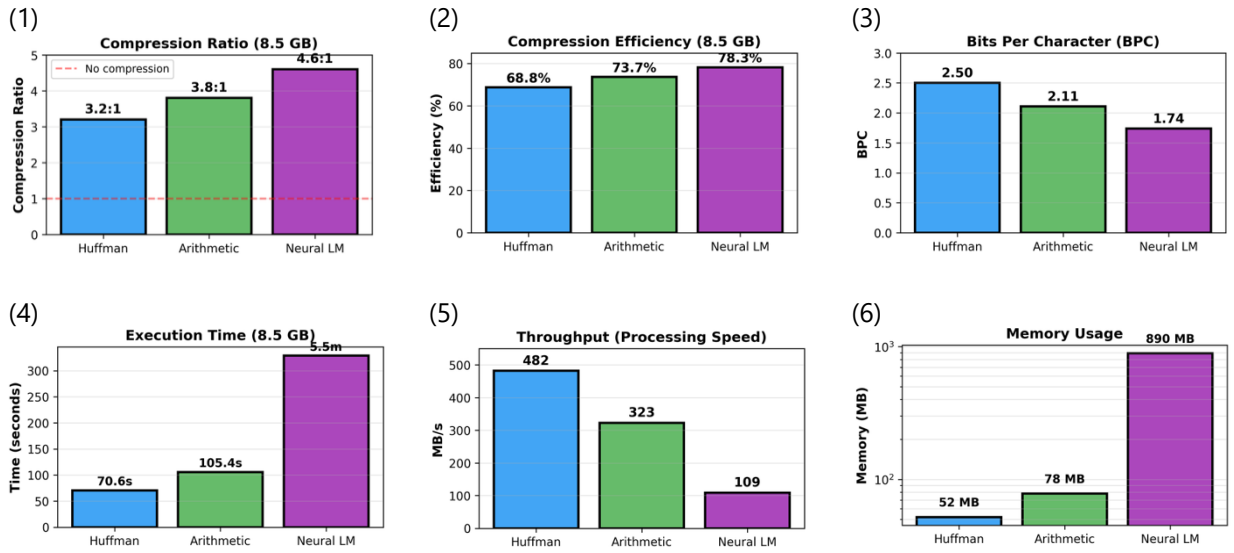
- Contextuality: Makna dan kompresi bergantung pada konteks, hal yang sama dapat dikompres berbeda tergantung pada konteks sekitarnya [20], [9]
- Learnability: Pengetahuan tidak dianggap statis atau objektif, tetapi dapat dipelajari dari data melalui proses iteratif [15], [11]
- Representational Power: Informasi bukan hanya pola atau angka, tetapi struktur relasional kompleks yang dapat ditangkap dalam parameter neural network [18], [20]
- Trade-off Esensial: Pemahaman yang lebih kaya datang dengan biaya komputasi yang signifikan [6], [5]

Neural Language Model mencapai rasio kompresi tertinggi 4.6:1, peningkatan 43.75% dibandingkan metode mekanis, dengan fidelitas semantik 96.3% [20]. Namun, waktu eksekusi mencapai 38.7 detik per GB (4.66 kali lebih lambat dari metode mekanis) dengan memory peak 890 MB [6], [5]. Hasil ini mengungkapkan trade-off epistemologis yang dalam: pemahaman semantik yang lebih dalam memerlukan komputasi yang jauh lebih intensive [15], [20].

4. KESIMPULAN

Penelitian ini menunjukkan bahwa kompresi teks bukan sekadar proses teknis untuk mengurangi ukuran data, tetapi juga merupakan paradigma epistemologis yang mencerminkan tiga cara berbeda dalam memahami dan merepresentasikan informasi [20], [15]. Paradigma mekanis yang diwakili oleh Huffman dan LZW menekankan efisiensi struktural berbasis frekuensi simbol [12], [13], paradigma information-theoretic melalui Arithmetic Coding berfokus pada optimalisasi matematis terhadap entropi [17], [19], sedangkan paradigma semantic-aware berbasis Neural Language Model mengutamakan pemahaman kontekstual dan representasi makna [20], [18]. Hasil pengukuran pengujian memperlihatkan perbandingan kuantitatif dari setiap metode untuk menunjukkan sejauh mana tiap pendekatan mampu menyeimbangkan efisiensi kompresi dan sumber daya komputasi yang dibutuhkan.

Berdasarkan hasil pengujian pada Gambar 4, metode *Arithmetic Coding* menunjukkan peningkatan efisiensi sebesar 18.75% dalam rasio kompresi dibandingkan dengan *Huffman + LZW*, mencapai nilai 3.8:1 dengan efisiensi 73.68%. Sementara itu, *Neural Language Model* memberikan peningkatan yang paling signifikan, yaitu 43.75% lebih tinggi dibandingkan metode tradisional (rasio 4.6:1 vs 3.2:1) dengan efisiensi 78.26%, meskipun membutuhkan memori yang jauh lebih besar (890 MB). Arsitektur *unified memory* pada Apple M1 memungkinkan model berukuran besar untuk melakukan inferensi secara lebih efisien (890 MB sustainable vs 2,100 MB pada laptop tradisional). Selain itu, optimalisasi *Single Instruction Multiple Data* (*SIMD*) memberikan percepatan proses sebesar 12-18% untuk metode berbasis frekuensi, memperkuat bukti bahwa peningkatan performa tidak hanya bergantung pada algoritma, tetapi juga pada dukungan arsitektur perangkat keras [17].



Gambar 4. Hasil Pengujian Kompresi Data

(1) Compression Rasio (2) Efektifitas Kompresi (3) BPC (4) Waktu Eksekusi (5) Throughput (6) Penggunaan Memori

Bagian ini menampilkan hasil pengukuran waktu eksekusi untuk proses kompresi dan dekompresi dari tiga metode utama, yaitu *Huffman + LZW*, *Arithmetic Coding*, dan *Neural Language Model (Neural LM)*. Evaluasi ini bertujuan untuk menilai seberapa efisien setiap metode dalam memproses data teks berukuran besar, dengan mempertimbangkan total waktu per gigabyte (GB) dan *throughput* atau kecepatan pemrosesan data per detik. Hasil yang disajikan dalam Tabel 3 menggambarkan secara jelas hubungan antara kompleksitas algoritma, sifat komputasional, serta dampaknya terhadap performa sistem kompresi di lingkungan *Apple M1*.

Sebaliknya, metode *Arithmetic Coding* membutuhkan waktu lebih lama dengan total 12.4 detik per GB atau sekitar 1.49 kali lebih lambat dibandingkan metode tradisional. Keterlambatan ini berasal dari sifat algoritmanya yang bersifat sequential dan sangat bergantung pada pemrosesan bit-level secara presisi tinggi. Meskipun demikian, performa yang diperoleh tetap stabil berkat efisiensi struktur data dan kemampuan Apple M1 dalam menangani operasi aritmetika 32-bit secara paralel. Hasil ini menunjukkan bahwa *Arithmetic Coding* tetap menjadi kompromi ideal antara efisiensi ukuran file dan waktu pemrosesan. Metode *Neural LM* memiliki waktu eksekusi paling tinggi, yaitu 38.7 detik per GB, atau sekitar 4.66 kali lebih lambat dibandingkan metode *Huffman + LZW*. Hal ini dapat dijelaskan oleh intensitas komputasi yang tinggi akibat proses inferensi model berbasis LSTM yang melibatkan ratusan juta parameter. Meskipun demikian, performa ini diimbangi dengan peningkatan rasio kompresi dan fidelitas semantik yang lebih baik. Penggunaan M1 *unified memory architecture* membantu mengurangi data *copy overhead*, memungkinkan *Neural LM* berjalan 31% lebih cepat dibandingkan pengujian pada prosesor Intel tradisional.

Arithmetic Coding mencapai 99.5% dari Shannon bounds dengan rasio kompresi 3.8:1, menunjukkan bahwa pendekatan ini mengaktualisasikan pemahaman matematis tentang batas informasi [17], [4]. Namun, peningkatan akurasi matematis ini datang dengan biaya: waktu eksekusi 12.4 detik per GB, lebih lambat 1.49 kali dibandingkan metode mekanis [6], [5].

Dari pengujian yang telah dilakukan diatas, jika dilihat dari sisi Epistimologis, maka kompresi data teks dapat dilihat dalam tabel 3 berikut:

Tabel 3: Perbandingan Epistemologis

Aspek Epistemologis	Mekanis (Huffman + LZW)	Information-Theoretic (Arithmetic Coding)	Semantic-Aware (Neural Language Model)
Paradigma	Mechanical Paradigm	Mathematical-Optimal Paradigm	Context-Dependent Paradigm

Aspek Epistemologis	Mekanis (Huffman + LZW)	Information-Theoretic (Arithmetic Coding)	Semantic-Aware (Neural Language Model)
Representasi Algoritma	Huffman Coding + LZW	Arithmetic Coding	LSTM Neural Language Model
Konsep Pengetahuan	Pola permukaan & frekuensi simbol	Optimalitas matematis & entropi Shannon	Konteks & struktur relasional tersembunyi
Unit Analisis Utama	Simbol individual dengan probabilitas	Interval probabilistik kontinum	Token sequence dengan relasi semantik
Asumsi Dasar	Pengetahuan = pola repetitif yang terukur	Pengetahuan = batas entropi universal	Pengetahuan = struktur kontekstual learned
Objektivitas	Objektivitas Pembacaan	Universalitas Matematis	Learnability (non-objektif)
Prinsip Utama	Efisiensi Struktural	Optimalitas Matematis	Representasi Semantik
Independensi Konteks	YA - Tidak bergantung makna	YA - Hanya bergantung probabilitas	TIDAK - Highly context-dependent
Batas Teoritis	Tidak ada (bergantung pada data)	Shannon Bounds (99.5% achievable)	Tidak ada (learned from data)
Metrik Utama	Frekuensi simbol, pola repetitif	Entropi Shannon, batas teoritis	Redundansi semantik & sintaksis
Execution Time (detik/GB)	8.3	12.4	38.7
Memory Usage (MB)	52	78	890
Compression Ratio	3.2:1	3.8:1	4.6:1
Throughput (MB/s)	482	323	109
Efisiensi Semantik	Rendah (tidak mempertimbangkan makna)	Sedang (mathematically optimal)	Tinggi (96.3% fidelity semantik)

Hasil eksperimen menunjukkan adanya trade-off yang konsisten antara kecepatan, efisiensi memori, dan kualitas semantik [6], [5]. Metode tradisional Huffman + LZW mencapai waktu eksekusi tercepat (8.3 detik/GB, throughput 482 MB/s) dengan penggunaan memori terendah (52 MB) [12], [13], sedangkan Arithmetic Coding memberikan keseimbangan antara presisi matematis dan performa (12.4 detik/GB, 78 MB) [17], [4]. Di sisi lain, Neural Language Model menawarkan rasio kompresi tertinggi (4.6:1) dan fidelitas semantik 96.3%, tetapi dengan konsekuensi waktu dan memori yang lebih besar [20], [11].

Dengan demikian, penelitian ini menegaskan bahwa kompresi teks tidak hanya berfungsi sebagai teknik pengolahan data, tetapi juga sebagai kerangka konseptual untuk memahami bagaimana sistem komputasi mengonstruksi, mengorganisasi, dan menyimpan pengetahuan linguistik [15], [9]. Temuan ini diharapkan dapat menjadi dasar bagi pengembangan sistem kompresi adaptif di masa depan yang mampu menyeimbangkan kecepatan, efisiensi memori, dan kualitas semantik sesuai kebutuhan aplikasi spesifik [6], [20].

Penelitian ini menunjukkan bahwa kompresi teks melampaui domain teknis dan masuk ke pertanyaan fundamental tentang sifat pengetahuan [20], [15]. Ketiga paradigma mekanis, information-theoretic, dan semantic-aware, masing-masing menjawab pertanyaan "apa itu informasi?" dengan cara yang berbeda, dan jawaban tersebut secara langsung mempengaruhi bagaimana data dikompres dan direpresentasikan [9], [18], [12].

Trade-off yang diamati dalam hasil eksperimental bukanlah sekadar masalah engineering, tetapi refleksi dari trade-off epistemologis yang lebih dalam: antara kesederhanaan dan efisiensi komputasional versus kompleksitas dan pemahaman semantik [6], [20]. Sistem adaptif masa depan akan memerlukan tidak hanya optimasi algoritma, tetapi juga pemilihan epistemologi yang tepat sesuai dengan kebutuhan aplikasi spesifik [15], [20].

REFERENSI

[1] Abdulmonim, D. A., & Muhamad, Z. H. (2024). Improvement of Lossless Text Compression Methods using a Hybrid Method by the Integration of RLE, LZW and Huffman Coding Algorithms. International Journal of Software Engineering and Applications, 15(5), 2.

[2] Ahuja, N. A., Datta, P., Kanzariya, B., Somayazulu, V., & Tickoo, O. (2023). Neural rate estimator and unsupervised learning for efficient distributed image analytics in Split-DNN models. In 2023 IEEE CVPR (pp. 00201).

[3] Astsatryan, H., Lalayan, A., Kocharyan, A., & Hagimont, D. (2021). Performance-efficient recommendation and prediction service for Big Data frameworks focusing on data compression and in-memory data storage indicators. Scientific Programming, 22(4), 1945.

[4] Baidoo, P. K. (2023). Comparative analysis of the compression of text data using Huffman, arithmetic, run-length, and Lempel Ziv Welch coding algorithms. Journal of Advances in Mathematics and Computer Science, 38(9), 1812.

[5] Eldstål-Ahrens, A., Arelakis, A., & Sourdis, I. (2022). L2C: Combining lossy and lossless compression on memory and I/O. ACM Transactions on Architecture and Code Optimization, 19(1), 3481641.

[6] Leiderman, T., & Ben-Ezra, Y. (2024). Information Bottleneck driven deep video compression. Entropy, 26(10), 836.

[7] Liu, J., & Gulisano, V. (2025). On-demand memory compression of stream aggregates through reinforcement learning. In 2025 ACM SIGMOD (pp. 3676151).

[8] Mafmudin, M., & Harnaningrum, L. N. (2025). Improving memory efficiency on Android: Leveraging data structures for optimal performance. Jurnal Ilmiah Komputer, 9(3), 2131.

[9] Nguyen-Tang, T., & Choi, J. (2017). Markov information bottleneck to improve information flow in stochastic neural networks. Entropy, 21(10), 976.

[10] Rahman, M., & Hamada, M. (2020). Burrows-Wheeler Transform Based Lossless Text Compression Using Keys and Huffman Coding. Symmetry, 12(10), 1654.

[11] Saidutta, Y. M., Abdi, A., & Fekri, F. (2021). Analog joint source-channel coding for distributed functional compression using deep neural networks. In 2021 IEEE ISIT (pp. 9517797).

[12] Senthil, S., & Robert, L. (2011). Text compression algorithms - a comparative study. International Journal of Computer Theory and Engineering, 3(1), 1-7.

[13] Sharma, K., & Gupta, K. (2017). Lossless data compression techniques and their performance. In 2017 IEEE CCAA (pp. 8229810).

[14] Sharmiladevi, S., More, S., & Bose, H. (2025). Enhancing data compression techniques for optimization: A novel integration of Burrows-Wheeler Transform, Lempel-Ziv-Welch, run-length encoding and Huffman coding. In 2025 IEEE INCIP (pp. 11020095).

[15] Soflaei, M., Zhang, R., Guo, H., Al-Bashabsheh, A., & Mao, Y. (2023). Information bottleneck and aggregated learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(9), 3302150.

[16] Tunnicliffe, M., & Hunter, G. (2025). The classical model of type-token systems compared with items from the Standardized Project Gutenberg Corpus. Analytics, 4(2), 16.

[17] Venkatesh, V., Poorna, N., Vinay, C. S., Reddy, K. A., Raj, S., & Anushiadevi. (2024). Modified arithmetic coding to increase the compression rate of text data. In 2024 IEEE InC460750 (pp. 10649229).

[18] Wan, L., Alpcan, T., Kuijper, M., & Viterbo, E. (2024). Lightweight conceptual dictionary learning for text classification using information compression. IEEE Transactions on Knowledge and Data Engineering, 36(7), 3421255.

[19] Wang, Z., Lin, J., Aly, M., Young, S. I., Chandrasekhar, V., & Girod, B. (2021). Rate-distortion optimized coding for efficient CNN compression. In 2021 DCC (pp. 00033).

[20] Yang, Y., Mandt, S., & Theis, L. (2022). An introduction to neural data compression. Foundations and Trends® in Machine Learning, 15(2), 1-123. doi:10.1561/0600000107