

INTEGRASI KOMPRESI BROTLI DAN BASE64 UNTUK TRANSFER DATA TERENKRIPSI YANG DIOPTIMALKAN DALAM SISTEM KOMUNIKASI MILITER BERBASIS HTTP

Asep Suryanta, Pipit Dewi Arnesia

Magister Teknologi Informasi, Sekolah Tinggi Manajemen Informatika dan Komputer Jakarta STI&K
Jl. BRI Radio Dalam No.17 1, RT.14/RW.3, Gandaria Utara, Kec. Kebayoran Baru,
Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta
zenilybaz@gmail.com

ABSTRAK

Keamanan dan efisiensi transmisi data berbasis teks pada protokol rentan seperti HTTP dalam sistem komunikasi militer merupakan isu kritis, terutama pada perangkat dengan sumber daya terbatas. Namun, metode konvensional sering kali belum optimal dalam menyeimbangkan aspek keamanan, rasio kompresi, dan kompatibilitas protokol. Oleh karena itu, penelitian ini mengusulkan metode inovatif dengan mengintegrasikan algoritma kompresi adaptif Brotli dan pengkodean Base64 untuk meningkatkan kerahasiaan dan efisiensi bandwidth. Metode ini diuji secara eksperimental dan dibandingkan dengan algoritma standar seperti Zlib, Gzip, dan Deflate berdasarkan rasio kompresi, waktu pemrosesan, dan integritas data. Hasilnya menunjukkan bahwa kombinasi Brotli+Base64 mencapai rasio kompresi tertinggi hingga 70,32% pada data HTML, secara konsisten mengungguli algoritma pembanding (49-65%). Meski waktu kompresi Brotli relatif lebih tinggi (hingga 0,382 detik), reduksi ukuran data yang signifikan dan kompatibilitas protokol bawaan menjadikannya kompromi yang efektif untuk komunikasi yang aman dan hemat bandwidth. Temuan ini memberikan kontribusi praktis bagi pengembangan sistem komunikasi militer yang tangguh dengan menyajikan metode yang secara simultan mengoptimalkan keamanan dan kecepatan transmisi.

Kata kunci : Kompresi Data, Brotli, Pengkodean Base64, Protokol HTTP.

1. PENDAHULUAN

Komunikasi data di lingkungan militer menuntut standar kerahasiaan, kecepatan, dan efisiensi yang sangat ketat. Dalam praktiknya, protokol berbasis teks seperti HTTP masih banyak digunakan, meskipun memiliki kerentanan signifikan terhadap penyadapan yang membahayakan informasi strategis [1], [2]. Solusi keamanan seperti HTTPS (HTTP Secure) telah diterapkan [3], namun tidak sepenuhnya kebal terhadap serangan seperti Man-in-the-Middle (MitM) [4], [5] dan hanya melindungi data selama transit, bukan pada saat penyimpanan [6], [7]. Implementasi enkripsi ujung-ke-ujung (E2EE) yang lebih kuat, meski meningkatkan keamanan, justru menimbulkan masalah baru: peningkatan ukuran data yang signifikan. Hal ini menyebabkan transmisi lebih lambat dan konsumsi bandwidth yang lebih tinggi [8], menciptakan konflik langsung antara tuntutan keamanan dan efisiensi operasional sebuah dilema kritis dalam konteks militer di mana kedua aspek tersebut sangat vital dan sumber daya seringkali terbatas.

Tantangan utama yang dihadapi adalah dikotomi antara keamanan dan efisiensi bandwidth dalam transmisi data militer terenkripsi. Teknik kompresi konvensional, seperti Transformasi Burrows-Wheeler (BWT) yang dikombinasikan dengan Pengkodean Panjang Berulang (RLE) [9], [10] terbukti kurang optimal untuk data teks militer yang dinamis dan beragam. Teknik-teknik ini seringkali menambah kompleksitas komputasi tanpa memberikan rasio kompresi yang memadai [11]. Dengan demikian,

terdapat kesenjangan penelitian untuk mengembangkan metode kompresi adaptif yang dapat secara signifikan mengurangi ukuran data teks terenkripsi tanpa membebani sumber daya komputasi perangkat yang terbatas [11], [12] [13].

Berdasarkan latar belakang dan identifikasi masalah di atas, tujuan utama penelitian ini adalah mengusulkan dan menguji metode transmisi data inovatif yang mengintegrasikan algoritma kompresi adaptif Brotli dengan pengkodean Base64. Tujuannya adalah untuk meningkatkan efisiensi bandwidth sekaligus menjaga keamanan dan kompatibilitas protokol. Secara spesifik, penelitian ini bertujuan untuk: Merancang dan mengimplementasikan model integrasi Brotli-Base64 yang kompatibel dengan protokol berbasis teks seperti HTTP, Mengevaluasi kinerja model dengan metrik rasio kompresi, waktu komputasi, kecepatan transmisi, dan konsumsi bandwidth, Membandingkan kinerja metode yang diusulkan dengan algoritma kompresi konvensional (Lzib, Gzip, Deflate), Menganalisis dampak sistem terhadap aspek keamanan dan menilai kelayakan implementasinya pada perangkat militer dengan sumber daya terbatas.

Untuk mengatasi kesenjangan tersebut, penelitian ini mengusulkan integrasi novel antara algoritma kompresi Brotli dan pengkodean Base64. Kebaruan utama terletak pada penerapan kombinasi ini yang disesuaikan secara spesifik untuk menyelesaikan pertukaran (trade-off) efisiensi-keamanan dalam komunikasi militer. Brotli dipilih karena kemampuannya mencapai rasio kompresi tinggi

(hingga ~70% untuk data teks) dan adaptabilitasnya yang dinamis, mengungguli standar lama seperti Gzip atau Deflate [13]. Namun, output biner dari Brotli tidak kompatibel secara native dengan saluran berbasis teks. Oleh karena itu, pengkodean Base64 diintegrasikan sebagai lapisan tambahan untuk mengonversi data terkompresi menjadi format teks ASCII yang aman, sehingga memastikan transmisi lancar melalui protokol seperti HTTP tanpa mengorbankan manfaat kompresi.

Rencana penyelesaian ini dijalankan melalui metodologi eksperimental. Kinerja sistem akan diuji pada berbagai sampel data teks militer simulatif. Metrik utama yang diukur meliputi rasio kompresi, waktu pemrosesan (kompresi/dekompresi), kecepatan transmisi, dan integritas data. Evaluasi komparatif akan dilakukan terhadap algoritma standar. Hasil eksperimen diharapkan dapat membuktikan hipotesis bahwa metode Brotli-Base64 mampu mengurangi ukuran data terenkripsi secara substansial (melebihi 30%) sambil menjaga kompatibilitas, keamanan, dan efisiensi komputasi yang layak untuk diterapkan dalam infrastruktur komunikasi militer.

2. TINJAUAN PUSTAKA

2.1. Tantangan Keamanan dan Efisiensi dalam Komunikasi Militer.

Komunikasi militer modern memerlukan keseimbangan kritis antara keamanan (kerahasiaan, integritas, ketersediaan) dan efisiensi bandwidth. Protokol berbasis teks seperti HTTP rentan terhadap penyadapan, sementara solusi seperti HTTPS—meski lebih aman dengan enkripsi TLS—terutama melindungi data in-transit dan tetap berisiko terhadap serangan seperti Man-in-the-Middle (MitM) jika otentikasi sertifikat lemah [14]. Untuk keamanan menyeluruh, enkripsi ujung-ke-ujung (E2EE) dengan algoritma kuat seperti AES sering diterapkan [15]. Namun, pendekatan ini menghasilkan overhead data yang signifikan (seperti padding dan metadata), yang mengurangi kecepatan transmisi dan membebani bandwidth sumber daya yang sangat berharga dan terbatas dalam operasi militer. Oleh karena itu, muncul pertukaran mendasar (trade-off) antara tingkat keamanan tinggi dan efisiensi transmisi, yang menuntut solusi inovatif untuk mengatasinya.

2.2. Peran dan Tantangan Kompresi Data dalam Konteks Keamanan.

Kompresi data merupakan solusi kunci untuk mengoptimalkan bandwidth dengan mengurangi redundansi sebelum transmisi. Dalam konteks komunikasi aman, kompresi harus diterapkan sebelum enkripsi agar efektif. Algoritma konvensional seperti Deflate (yang menjadi dasar Zlib dan Gzip) telah menjadi standar, namun kurang optimal untuk data teks militer modern yang dinamis dan terstruktur (seperti XML/JSON). Algoritma ini seringkali menambah kompleksitas komputasi tanpa

memberikan pengurangan ukuran yang signifikan pada data berentropi tinggi [16], [17]. Oleh karena itu, dibutuhkan algoritma kompresi yang lebih adaptif dan canggih yang tidak hanya menawarkan rasio kompresi tinggi, tetapi juga efisien secara komputasi untuk dijalankan pada perangkat militer dengan sumber daya terbatas.

2.3. Algoritma Kompresi Brotli: Keunggulan dan Potensi.

Brotli adalah algoritma kompresi lossless modern yang dirancang khusus untuk data web (HTML, CSS, JavaScript) dan dikenal menghasilkan rasio kompresi 20-26% lebih baik daripada Gzip untuk data teks. Keunggulan ini dicapai melalui model kontekstual tingkat dua dan kamus statis yang besar berisi string umum. Brotli juga adaptif dengan 11 tingkat kompresi untuk menyeimbangkan ukuran dan kecepatan [18], [19]. Namun, outputnya berupa data biner, yang memerlukan pengkodean tambahan untuk kompatibilitas dengan saluran berbasis teks

2.4. Gzip, Zlib dan Deflate

Gzip dan Zlib adalah implementasi luas dari algoritma Deflate, yang menggabungkan LZ77 dan pengkodean Huffman. Deflate menawarkan keseimbangan yang baik antara rasio kompresi dan kecepatan, menjadikannya standar de facto untuk kompresi web dan banyak format arsip (seperti ZIP, PNG) [20], [21], [22]. Gzip, yang menggunakan Zlib, dapat mengurangi ukuran data teks hingga 70% dan didukung hampir universal. Namun, untuk data yang sangat terstruktur dan dinamis, kinerjanya dapat kalah dibandingkan algoritma yang lebih baru seperti Brotli [23].

2.5. Pengkodean Base64 sebagai Enabler Kompatibilitas.

Pengkodean Base64 adalah skema konversi biner-ke-teks yang mengubah data biner menjadi string karakter ASCII yang aman untuk transmisi melalui protokol berbasis teks seperti HTTP atau email [20], [21]. Skema ini sangat penting sebagai jembatan kompatibilitas, memungkinkan output biner dari algoritma kompresi (seperti Brotli) untuk dikirimkan tanpa korupsi melalui saluran yang hanya mendukung teks. Meskipun proses ini menambah overhead ukuran sekitar 33% [23], pengurangan ukuran signifikan dari kompresi yang efisien diharapkan dapat mengimbangnya, sehingga tetap menghasilkan ukuran akhir yang lebih kecil dibandingkan data asli [24], [25], [26], [27], [28], [29], [30], [31], [32].

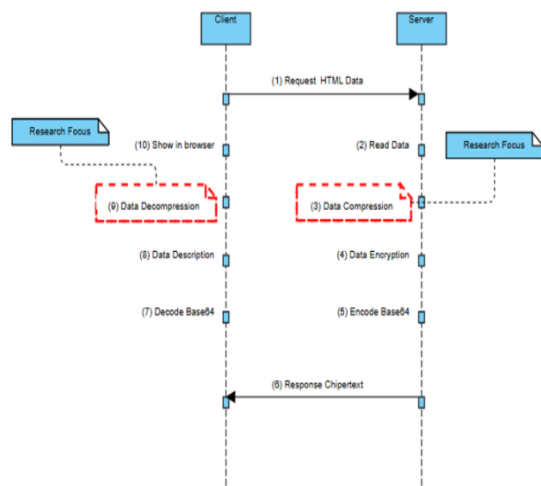
2.6. Penelitian Terdahulu dan Posisi Penelitian Ini.

Berdasarkan kajian pustaka, terdapat celah penelitian (research gap) dalam kurangnya solusi terpadu yang secara simultan mengoptimalkan efisiensi bandwidth dan keamanan untuk komunikasi militer berbasis teks. Penelitian terdahulu [9] telah

mengidentifikasi trade-off antara enkripsi kuat dan kinerja sistem, serta merekomendasikan integrasi teknik kompresi, namun belum mengeksplorasi implementasi spesifik dengan algoritma modern. Sementara itu, studi [12] menyoroti perlunya sistem enkripsi yang lebih efisien dalam konteks militer. Algoritma kompresi klasik (seperti BWT+RLE) memiliki kompleksitas komputasi tinggi dan kinerja tidak konsisten untuk data militer yang dinamis [11]. Di sisi lain, algoritma modern seperti Brotli yang unggul dalam rasio kompresi untuk data teks [13], belum diuji secara mendalam dalam konteks ini, terutama ketika dikombinasikan dengan encoding Base64 untuk memastikan kompatibilitas. Oleh karena itu, penelitian ini mengusulkan dan menguji pipeline terintegrasi Brotli + Base64 sebagai solusi holistik. Posisi kebaruan penelitian ini adalah dalam penerapan dan evaluasi kombinasi khusus ini untuk secara langsung mengatasi pertukaran efisiensi-keamanan dalam komunikasi militer, dengan target mencapai pengurangan ukuran data yang substansial (>30%) sambil menjaga kompatibilitas protokol dan kelayakan implementasi pada perangkat berdaya terbatas.

3. METODE PENELITIAN

Penelitian ini menggunakan metode eksperimental kuantitatif untuk menguji pengaruh kombinasi algoritma kompresi teks terhadap efisiensi dan keamanan komunikasi data militer melalui HTTP. Pendekatan ini memungkinkan pengendalian variabel kunci serta pengukuran empiris terhadap parameter kuantitatif seperti rasio kompresi, waktu transmisi, dan keberhasilan enkripsi, sekaligus menguji hipotesis yang diajukan.



Gambar 1. Diagram Sequence proses kompres

Diagram sequence penelitian ini menggambarkan alur komunikasi lengkap. Klien mengirimkan Permintaan Data HTML (1) ke Server. Di sisi Server, data HTML dibaca (2) lalu menjalani tiga proses inti secara berurutan: kompresi (3) (dengan algoritma seperti Brotli/Zlib/Gzip/Deflate), enkripsi (4), dan encoding Base64 (5). Data hasil proses ini

kemudian dikirim sebagai Respons Teks Sandi (6) ke Klien. Di sisi Klien, data didekode dari Base64 (7), didekripsi (8), dan didekompresi (9) untuk mengembalikan HTML asli, yang akhirnya ditampilkan di browser (10).

3.1. Pengaturan dan Konfigurasi Eksperimental.

Dataset yang digunakan dalam pengujian terdiri dari 10 laporan operasi bergaya militer dalam format HTML dengan berbagai ukuran, sekitar 100, 200, 300, 400, 500, 600, 700, 800, 900, dan 1000 KB, untuk mewakili volume data realistis dalam lingkungan taktis. Prototipe sistem diimplementasikan menggunakan Python 3.11 dengan memanfaatkan pustaka kompresi Brotli, Zlib, Gzip, dan Deflate, serta pengkodean Base64. Algoritma Brotli diuji di seluruh rentang tingkat kompresinya (1 hingga 11), dengan level 6 sebagai konfigurasi default untuk analisis perbandingan utama karena mewakili pengaturan awal yang seimbang antara kecepatan dan rasio kompresi. Semua pengujian dilakukan dalam lingkungan perangkat keras yang terkontrol, menggunakan komputer dengan prosesor Intel Core i5-1135G7 dan RAM 8 GB, untuk memastikan konsistensi dan validitas pengukuran kinerja komputasi.

3.2. Prosedur Eksperimental dan Variabel.

Eksperimen terstruktur dalam empat fase: (1) Pengukuran Garis Dasar untuk memperoleh ukuran dan waktu transmisi data terenkripsi tanpa kompresi; (2) Evaluasi Model yang Diusulkan dengan mengukur kinerja metode Brotli + Base64 terhadap metrik ukuran data, waktu kompresi/dekompresi, dan rasio kompresi; (3) Analisis Komparatif dengan membandingkan kinerja tiga algoritma benchmark (Zlib, Gzip, Deflate) yang masing-masing dikombinasikan dengan Base64 menggunakan metrik yang sama; serta (4) Analisis Data untuk mengumpulkan dan mengevaluasi seluruh hasil, dengan fokus utama pada parameter Rasio Kompresi sebagai ukuran efektivitas setiap algoritma.

$$\text{Compression efficiency}(C) = \frac{S_{in} - S_{out}}{S_{in}} \times 100\%$$

- Overhead Komputasi: diukur melalui waktu kompresi dan dekompresi (dalam milidetik).
- Efisiensi Transmisi: diperoleh dari ukuran payload akhir paket HTTP.

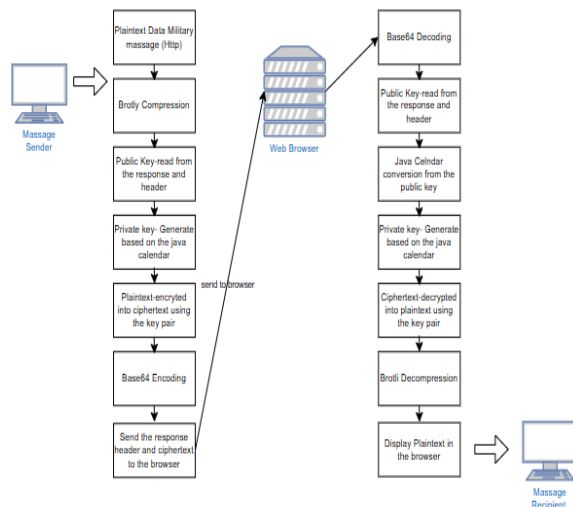
Efisiensi kompresi (CC) dihitung sebagai persentase pengurangan ukuran data setelah kompresi. Rumusnya adalah:

$$CC = [(S_{in} - S_{out}) / S_{in}] \times 100\%,$$

di mana S_{in} adalah ukuran data asli dan S_{out} adalah ukuran setelah kompresi. Semakin tinggi nilai CC, semakin efektif algoritma dalam mengurangi redundansi data.

3.3. Desain Program Kompresi..

Diagram desain program kompresi dalam penelitian ini adalah sebagai berikut:



Gambar 2. Proses Enkripsi dan Proses Dekripsi

Proses Enkripsi: Data asli dikompresi dengan Brotli (lossless) untuk efisiensi bandwidth, lalu dienkripsi menggunakan kunci publik dari header respons. Kunci pribadi dihasilkan melalui fungsi hashing berbasis kalender Jawa, menambah lapisan keamanan unik. Data terenkripsi kemudian dikodekan menjadi Base64 agar kompatibel dengan HTTP, sebelum dikirim ke browser bersama header respons.

Proses Dekripsi: Di sisi penerima, data didekode dari Base64 ke biner. Kunci publik diekstraksi dari header respons, dikonversi ke parameter kalender Jawa, lalu digunakan untuk menghasilkan kunci pribadi melalui fungsi hashing yang sama. Dengan kunci pribadi ini, data didekripsi, didekompresi, dan hasil akhir ditampilkan di browser.

3.4. Kompresi Brotly

Brotli adalah algoritma kompresi lossless dari Google yang menghasilkan rasio kompresi tinggi dengan kecepatan dekompresi efisien, khususnya optimal untuk data teks. Algoritma ini dirancang untuk transfer web (HTTP) dan kompresi generik, mengandalkan tiga komponen inti yang bekerja sinergis: kamus statis, pencocokan LZ77, dan pengkodean entropi Huffman.

3.5. Kamus Statis (Static Dictionary).

Brotli menggunakan kamus statis pradeфинisi yang berisi ribuan kata, frasa, dan fragmen umum (terutama dalam konteks web dan bahasa Inggris). Kamus ini terintegrasi langsung ke dalam algoritma, sehingga tidak perlu dikirim bersama data. Ketika urutan byte dalam data masukan cocok dengan entri kamus, urutan tersebut digantikan dengan indeks kamus yang ringkas, yang secara signifikan meningkatkan efisiensi kompresi untuk kata dan pola yang sering muncul [33]. Efektivitas kamus ini bersifat selektif, tergantung pada karakteristik linguistik dan

kontekstual data misalnya, kata-kata seperti "compression", "data", "Google", dan "lossless" yang umum dalam domain teknologi secara konsisten dikompresi dengan sangat efisien karena telah terdaftar dalam kamus bawaan.

Tabel 1. Kamus Statis Brotly

Kompresi	✓ ya (compression)	indeks kamus Brotli
data	✓ ya	indeks kamus Brotli
Google	✓ ya	indeks kamus Brotli
lossless	✓ ya	indeks kamus Brotli
algoritma	✗ tidak (tidak umum global)	disimpan literal biasa
yang	✗ tidak (kata bahasa Indonesia)	disimpan literal biasa

3.6. Encode Base64.

Base64 merupakan skema pengkodean biner-ke-tekst (binary-to-text encoding) yang mengonversi data biner menjadi representasi teks menggunakan 64 karakter ASCII yang dapat dicetak (printable characters). Teknik ini dikembangkan untuk memungkinkan transmisi data biner yang aman melalui saluran komunikasi yang dirancang khusus untuk data teks, seperti protokol email (MIME), XML, JSON, atau protokol web lainnya yang hanya mendukung karakter teks.

Secara fundamental, Base64 mengatasi masalah inkompatibilitas antara data biner dan sistem yang hanya memproses teks dengan mentransformasi setiap 3 byte (24 bit) data biner menjadi 4 karakter ASCII 6-bit, sehingga menghasilkan rasio ekspansi sekitar 33% (data terencode menjadi 4/3 atau sekitar 133% dari ukuran asli).

Algoritma Base64 menggunakan himpunan 64 karakter utama yang terdiri dari:

- 26 huruf besar: A-Z
- 26 huruf kecil: a-z
- 10 angka: 0-9
- simbol: + dan /

Selain itu, karakter '=' digunakan sebagai padding di akhir data untuk memastikan panjang kelipatan 4. Karakter-karakter ini dipilih karena universal tersedia dan tidak memiliki makna khusus dalam kebanyakan protokol teks.

Tabel 2. menunjukkan indeks karakter Base64 beserta representasi biner 6-bit-nya:

Indeks	Karakter	Kode Biner (6-bit)	Indeks	Karakter	Kode Biner (6-bit)
0	A	000000	32	g	100000
1	B	000001	33	h	100001
...	...	...	...	...	...
25	Z	011001	57	5	111001
26	a	011010	58	6	111010
27	b	011011	59	7	111011
28	c	011100	60	8	111100
29	d	011101	61	9	111101
30	e	011110	62	+	111110
31	f	011111	63	/	111111

4. HASIL DAN PEMBAHASAN

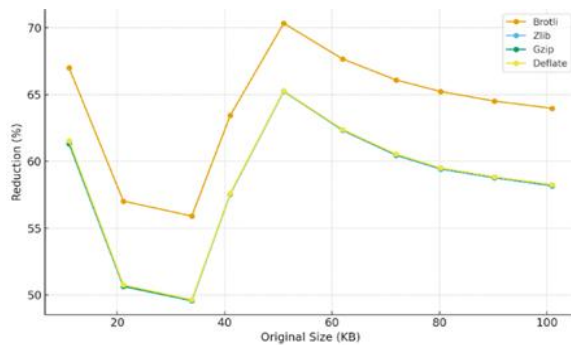
Penelitian ini merancang dan mengimplementasikan jalur pengujian terstruktur yang otomatis untuk mengevaluasi kinerja algoritma kompresi dalam simulasi sistem komunikasi taktis. Sistem ini mensimulasikan alur kerja pertukaran data dunia nyata, mencakup seluruh proses mulai dari persiapan muatan, kompresi, hingga enkripsi untuk transmisi.

Uji coba menggunakan konten HTML yang merepresentasikan pesan atau laporan situasional,

yang dibuat secara terprogram. Sepuluh varian data dengan ukuran 10 KB hingga 100 KB (interval ~10 KB) digunakan untuk mencakup spektrum beban data operasional umum. Selain itu, dilakukan uji ekstrem dengan menyisipkan gambar simulasi untuk menghasilkan sampel besar (186 KB hingga 1497 KB). Seluruh data divalidasi terlebih dahulu untuk memastikan struktur dan konsistensinya, guna menjaga validitas perbandingan antarkinerja algoritma.

Tabel 3. Hasil uji kompresi

Original Size (KB)	Algoritma	Size After Compression (KB)	Final Size (Base64) (KB)	Reduction vs Original	Compression Time (seconds)
11.03	Brotli	2.73	3.64	66.99%	0.213
	Zlib	3.19	4.26	61.36%	0.004
	Gzip	3.20	4.27	61.32%	0.001
	Deflate	3.18	4.24	61.54%	0.001
21.14	Brotli	6.81	9.09	57.02%	0.091
	Zlib	7.82	10.43	50.68%	0.003
	Gzip	7.82	10.43	50.64%	0.003
	Deflate	7.81	10.41	50.75%	0.002
33.88	Brotli	11.20	14.94	55.90%	0.390
	Zlib	12.80	17.07	49.61%	0.007
	Gzip	12.81	17.08	49.57%	0.004
	Deflate	12.79	17.06	49.64%	0.006
41.03	Brotli	11.25	15.01	63.42%	0.148
	Zlib	13.06	17.41	57.56%	0.003
	Gzip	13.06	17.42	57.55%	0.013
	Deflate	13.04	17.39	57.60%	0.003
50.99	Brotli	11.35	15.13	70.32%	0.150
	Zlib	13.28	17.71	65.27%	0.004
	Gzip	13.30	17.73	65.23%	0.006
	Deflate	13.28	17.71	65.27%	0.004
61.97	Brotli	15.03	20.05	67.65%	0.202
	Zlib	17.52	23.36	62.31%	0.014
	Gzip	17.50	23.34	62.34%	0.007
	Deflate	17.49	23.32	62.38%	0.010
71.92	Brotli	18.29	24.39	66.08%	0.249
	Zlib	21.34	28.45	60.44%	0.010
	Gzip	21.31	28.41	60.49%	0.018
	Deflate	21.29	28.39	60.53%	0.014
80.21	Brotli	20.92	27.89	65.22%	0.286
	Zlib	24.42	32.56	59.41%	0.009
	Gzip	24.38	32.51	59.47%	0.012
	Deflate	24.36	32.48	59.50%	0.010
90.22	Brotli	24.02	32.03	64.50%	0.364
	Zlib	27.91	37.21	58.76%	0.012
	Gzip	27.87	37.17	58.80%	0.014
	Deflate	27.86	37.14	58.83%	0.015
100.94	Brotli	27.28	36.38	63.96%	0.382
	Zlib	31.68	42.25	58.15%	0.011
	Gzip	31.62	42.17	58.22%	0.015
	Deflate	31.61	42.14	58.25%	0.029



Gambar 3. Grafik uji data teks HTML

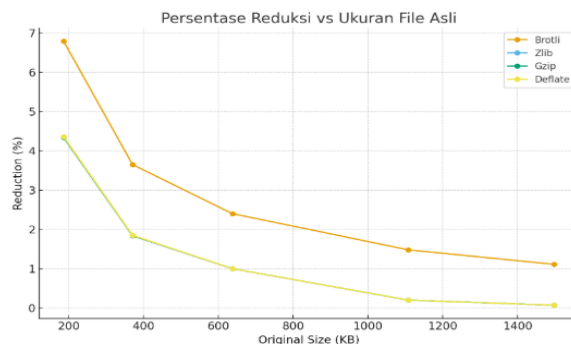
Grafik di atas membandingkan efektivitas kompresi dari empat algoritma, yaitu Brotli, Zlib, Gzip, dan Deflate, berdasarkan persentase pengurangan ukuran file relatif terhadap ukuran aslinya. Secara keseluruhan, Brotli secara konsisten memberikan hasil kompresi tertinggi di semua ukuran file, mencapai pengurangan maksimum sekitar 70% pada 50,99 KB, sementara tiga algoritma lainnya berkinerja relatif sama, dengan pengurangan berkisar

antara 49% hingga 62%. Seiring bertambahnya ukuran file asli, tingkat pengurangan untuk setiap algoritma cenderung mencapai titik jenuh, tetapi Brotli mempertahankan keunggulan signifikan dibandingkan Zlib, Gzip, dan Deflate, menunjukkan bahwa Brotli menawarkan efisiensi kompresi terbaik untuk rentang ukuran file yang diuji.

Sebagai bagian dari analisis komprehensif, pengujian ekstrem dilakukan yang melampaui skenario penelitian inti untuk mengevaluasi batas kemampuan algoritma kompresi. Pengujian ini dirancang untuk menguji respons algoritma terhadap skenario yang lebih kompleks dan Jenis muatan representatif, yaitu dokumen HTML yang berisi tidak hanya teks tetapi juga gambar dengan berbagai ukuran. Tujuan pengujian ini adalah untuk mengamati dan mengukur efektivitas kompresi dari empat algoritma Brotli, Zlib, Gzip, dan Deflate pada konten hibrida yang lebih mendekati kondisi pertukaran data dunia nyata, meskipun memiliki karakteristik yang berbeda dari data teks murni. Hasil pengujian untuk sampel data HTML dengan konten gambar disajikan secara rinci pada Tabel 4.2 di bawah ini.

Tabel 4. Hasil Tes Kompresi Teks dan Gambar

Original Size (KB)	Algoritma	Size After Compression (KB)	Final Size (Base64) (KB)	Reduction vs Original	Compression Time (seconds)
186.56	Brotli	130.42	173.89	6.79%	0.811
	Zlib	133.83	178.45	4.35%	0.018
	Gzip	133.84	178.45	4.34%	0.025
	Deflate	133.82	178.43	4.36%	0.019
370.8	Brotli	267.96	357.28	3.65%	1.804
	Zlib	272.97	363.96	1.84%	0.044
	Gzip	272.97	363.96	1.84%	0.046
	Deflate	272.95	363.94	1.85%	0.076
637.58	Brotli	466.69	622.25	2.40%	2.690
	Zlib	473.41	631.22	1.00%	0.098
	Gzip	473.41	631.22	1.00%	0.076
	Deflate	473.40	631.20	1.00%	0.071
1107.67	Brotli	818.42	1091.23	1.48%	4.832
	Zlib	829.11	1105.48	0.20%	0.193
	Gzip	829.11	1105.48	0.20%	0.170
	Deflate	829.09	1105.46	0.20%	0.124
1497.07	Brotli	1110.36	1480.48	1.11%	6.865
	Zlib	1123.59	1498.12	0.07%	0.288
	Gzip	1123.59	1498.12	0.07%	0.455
	Deflate	1123.57	1498.10	0.07%	0.168



Gambar 4. Grafik Uji Data Teks dan Gambar HTML

Grafik Persentase Pengurangan dengan Ukuran File Asli membandingkan efektivitas kompresi dari empat algoritma, yaitu Brotli, Zlib, Gzip, dan Deflate, berdasarkan persentase pengurangan ukuran file sebelum dan sesudah kompresi. Terlihat bahwa persentase pengurangan menurun secara signifikan seiring dengan peningkatan ukuran file asli, yang menunjukkan bahwa algoritma kompresi menjadi kurang efektif pada file berukuran besar. Brotli secara konsisten memberikan tingkat pengurangan terbaik, mencapai sekitar 6,79% pada ukuran file 186,56 KB, sedangkan Zlib, Gzip, dan Deflate berada dalam kisaran yang serupa, dimulai dari 4,36% dan menurun

dengan cepat hingga sekitar 0,07% pada ukuran file 1497,07 KB. Pola penurunan ini menunjukkan bahwa untuk ukuran data yang besar, keempat algoritma tersebut menghasilkan peningkatan ukuran yang sangat minimal, tetapi Brotli masih mempertahankan keunggulan relatif dibandingkan yang lain.

5. KESIMPULAN DAN SARAN

Berdasarkan analisis komparatif data pada kedua tabel pengujian, dapat disimpulkan bahwa terdapat trade-off mendasar antara efisiensi kompresi dan kecepatan eksekusi di antara keempat algoritma yang diuji. Untuk data teks HTML murni (10–100 KB), algoritma Brotli secara konsisten menghasilkan rasio kompresi tertinggi (55,9%–70,32%), tetapi dengan waktu kompresi yang jauh lebih lama (hingga 0,382 detik) dibandingkan dengan Zlib, Gzip, dan Deflate yang memiliki rasio kompresi lebih rendah (49,57%–65,27%) tetapi menyelesaikan proses dalam waktu kurang dari 30 milidetik. Sementara itu, untuk data teks dan gambar hibrida (186–1497 KB), efektivitas semua algoritma menurun drastis dengan rasio pengurangan di bawah 7%, di mana keunggulan Brotli menjadi praktis tidak signifikan, sedangkan algoritma berbasis Zlib mempertahankan kecepatan pemrosesan yang jauh lebih unggul. Temuan ini menunjukkan bahwa pemilihan algoritma harus mempertimbangkan prioritas sistem: Brotli cocok untuk optimasi ukuran pada data teks murni ketika waktu komputasi bukan kendala utama, sedangkan Zlib/Gzip/Deflate lebih sesuai untuk aplikasi waktu nyata atau data yang didominasi oleh konten multimedia yang telah dikompresi sebelumnya.

Oleh karena itu, penelitian selanjutnya sangat disarankan untuk mengembangkan dan menguji implementasi sistem kompresi adaptif yang mampu secara dinamis memilih algoritma terbaik berdasarkan analisis konten waktu nyata, sumber daya komputasi yang tersedia, dan kondisi jaringan saat ini. Sistem seperti itu diharapkan dapat secara otomatis beralih ke Brotli untuk muatan teks padat ketika bandwidth menjadi faktor pembatas, dan beralih ke keluarga algoritma Zlib/Gzip untuk data campuran atau multimedia ketika latensi menjadi prioritas utama, sehingga mencapai efisiensi transmisi optimal dalam berbagai skenario operasional militer yang dinamis.

DAFTAR PUSTAKA

- [1] A. Basuki, P. Manajemen Pertahanan Akademi Militer, I. Kusno Hadi, M. Raindra Prodi Manajemen Pertahanan Akademi Militer, And M. Irul Ferdiyansyah, “Ciptaan Disebarluaskan Di Bawah Lisensi Creative Commons Atribusi 4.0 Internasional,” Vol. 11, No. 2, Pp. 2355–5262, 2024.
- [2] K. A. McKay And D. A. Cooper, “Guidelines For The Selection, Configuration, And Use Of Transport Layer Security (Tls) Implementations,” Gaithersburg, Md, Aug. 2019. Doi: 10.6028/Nist.Sp.800-52r2.
- [3] Z. Meng *Et Al.*, “Achieving Consistent Low Latency For Wireless Real-Time Communications With The Shortest Control Loop,” In *Sigcomm 2022 - Proceedings Of The Acm Sigcomm 2022 Conference*, Association For Computing Machinery, Inc, Aug. 2022, Pp. 193–206. Doi: 10.1145/3544216.3544225.
- [4] Bogdan Barchuk, “Man-In-The-Middle (Mitm) Attacks: Techniques And Defenses,” *World Journal Of Advanced Engineering Technology And Sciences*, Vol. 13, No. 2, Pp. 919–933, Dec. 2024, Doi: 10.30574/Wjaets.2024.13.2.0650.
- [5] I. Olanrewaju Ibraheem And N. Olawale Ibrahim, “Analyzing Vulnerabilities In Network Protocols Using Wireshark: A Case Study On Http And Https,” 2024. [Online]. Available: <https://www.researchgate.net/publication/388026440>
- [6] D. Auliafitri, E. Rizkisuro, M. R. M. Malik, And A. Setiawan, “Optimalisasi Pengujian Penetrasi: Penerapan Serangan Mitm (Man In The Middle Attack) Menggunakan Websploit,” *Journal Of Internet And Software Engineering*, Vol. 1, No. 3, P. 12, Jun. 2024, Doi: 10.47134/Pjise.V1i3.2620.
- [7] J. Zhou, W. Fu, W. Hu, Z. Sun, T. He, And Z. Zhang, “Challenges And Advances In Analyzing Tls 1.3-Encrypted Traffic: A Comprehensive Survey,” Oct. 01, 2024, *Multidisciplinary Digital Publishing Institute (Mdpi)*. Doi: 10.3390/Electronics13204000.
- [8] A. Bhakti, A. Sudirman, R. Widya Setiabudi Sumadinata, And A. Bainus, “State Defense Strategy In Facing Cyber Threats After Hacking Incidents On Government Institutions: A Case Study In Indonesia,” *J. Hum. Secur.*, Vol. 20, No. 1, Pp. 109–117, 2024, Doi: 10.12924/Johs2024.20116.
- [9] M. Pasetti, E. Sisinni, P. Ferrari, P. Bellagente, And D. Zaninelli, “Comprehensive Evaluation Of Lossless Compression Algorithms In A Real Use Case For Smart Grid Applications,” *Sustainable Energy, Grids And Networks*, Vol. 36, Dec. 2023, Doi: 10.1016/J.Segan.2023.101238.
- [10] L. Major, A. Clare, J. W. Daykin, B. Mora, And C. Zarges, “Heuristics For The Run-Length Encoded Burrows–Wheeler Transform Alphabet Ordering Problem,” *Journal Of Heuristics*, Vol. 31, No. 1, Mar. 2025, Doi: 10.1007/S10732-025-09548-3.
- [11] S. Fiergolla And P. Wolf, “Improving Run Length Encoding By Preprocessing,” Mar. 2021, [Online]. Available: <http://arxiv.org/abs/2101.05329>
- [12] P. N. Swanzy, A. M. Abukari, And E. D. Ansong, “Data Security Framework For Protecting Data In Transit And Data At Rest In The Cloud,” *Current Journal Of Applied Science And*

- Technology, Vol. 43, No. 6, Pp. 61–77, May 2024, Doi: 10.9734/Cjast/2024/V43i64387.
- [13] E. H. Susanto, Y. Dwi Cahyono, And R. Budi, “Improving End-To-End Encryption Security On Http Using A Gregorian Dan Javanese Calendar-Based Key Generator,” 2025.
- [14] S. Alroomi And F. Li, “Measuring Website Password Creation Policies At Scale,” In *Ccs 2023 - Proceedings Of The 2023 Acm Sigsac Conference On Computer And Communications Security*, Association For Computing Machinery, Inc, Nov. 2023, Pp. 3108–3122. Doi: 10.1145/3576915.3623156.
- [15] M. Paquet-Clouston, S. O. Paquette, S. Garcia, And M. J. Erquiaga, “Entanglement: Cybercrime Connections Of A Public Forum Population,” *J. Cybersecur.*, Vol. 8, No. 1, 2022, Doi: 10.1093/Cybsec/Tyac010.
- [16] G. P. Tiwary, E. Stroulia, And A. Srivastava, “Compression Of Xml And Json Api Responses,” *Ieee Access*, Vol. 9, Pp. 57426–57439, 2021, Doi: 10.1109/Access.2021.3073041.
- [17] A. Brasoveanu, M. Moodie, And R. Agrawal, “Textual Evidence For The Perfunctoriness Of Independent Medical Reviews,” In *Ceur Workshop Proceedings*, Ceur-Ws, 2020, Pp. 1–9. Doi: 10.1145/Nnnnnnn.Nnnnnnn.
- [18] J. Alakuijala, A. Farruggia, P. Ferragina, E. Kliuchnikov, And R. Obryk, “Brotli: A General-Purpose Data Compressor,” 2021. [Online]. Available: <http://Caniuse.Com/#Search=Brotli>
- [19] Z. Ali Syed And T. Rahim Soomro, “Compression Algorithms: Brotli, Gzip And Zopfli Perspective,” *Indian J. Sci. Technol.*, Vol. 11, No. 45, Pp. 1–4, Dec. 2021, Doi: 10.17485/Ijst/2018/V11i45/117921.
- [20] M. Yiko Satriyawibawa, P. Nurtantio Andono, L. Way Soong, And N. Poh Kiat, “Lsb-2 Steganography With Brotli Compression And Base64 Encoding For Improving Data Embedding Capacity,” *Jurnal Dan Penelitian Teknik Informatika*, Vol. 8, No. 2, 2024, Doi: 10.33395/V8i2.13573.
- [21] A. Pinandito, A. Putra Kharisma, E. Muhammad, And A. Jonemaro, “Architectural Design Of Representational State Transfer Application Programming Interface With Application-Level Base64-Encoding And Zlib Data Compression,” 2023. [Online]. Available: Www.Jitecs.Ub.Ac.Id
- [22] D. Pradeka, Z. Khaerunnisa, S. Aqila Humaira, And A. Salsa Billa, “Digital Data Security Using A Combination Of Base64 Encoding, Rail Fence Cipher, And Gzip Compression Pradeka Et Al., Digital Data Security Using A Combination Of Base64 Encoding, Rail Fence Cipher ... [52],” *Coelite*, Vol. 4, No. 1, Pp. 51–60, 2025, Doi: 10.17509/Coelite.V4i1.82498.
- [23] F. Baso, “Analysis And Utilization Of The Base64 Algorithm For Image Encryption And Decryption Security In Web-Based Images,” *Computer, Information, Embedded, Network, And Intelligence System*, Vol. 1, No. 1, P. 2023, 2023, [Online]. Available: <https://Journal.Lontaradigitech.Com/Scientist>
- [24] Et Al. Rahman, “Serangan Man-In-The-Middle (Mitm) Di Jaringan Publik: Studi Dan Solusi Simulasi Serangan Password Cracking Menggunakan Hydra (Rahman, Et Al.),” 2025, Doi: 10.63822/Np7skj98.
- [25] E. H. Susanto, D. P. Pratama, And R. S. Nurpratama, “Optimizing Digital Image Steganography To Enhance The Security Of Secret Message Delivery,” *International Journal Of Engineering Continuity*, Vol. 3, No. 1, Pp. 38–58, Mar. 2024, Doi: 10.58291/Ijec.V3i1.194.
- [26] T. Kandaga, “Analisis Penerapan Kompresi Dan Dekompresi Data Dengan Menggunakan Metode Statistik Dan Kamus,” 2022.
- [27] R. Saputra, Y. Reswan, And Y. Darmi, “Fast Method Of Image File Compression In Web-Based Application Form Using Huffman Algorithm Metode Cepat Kompresi File Citra Pada Form Aplikasi Berbasis Web Menggunakan Algoritma Huffman,” *Jurnal Komitek*, Vol. 3, No. 2, Pp. 285–294, 2023, Doi: 10.53697/Jkomitek.V3i2.
- [28] Aswar Hanif, “Komparasi Performa Algoritma Kompresi Data Lossless Menggunakan Rasio Kompresi Dan Penghematan Ruang,” *J-Intech (Journal Of Information And Technology)*, 2022.
- [29] M. C. Aruan And Wanti Rahayu, “Analisis Performa Algoritma Kompresi Data Dalam Penyimpanan Dan Transfer Data,” *Lancah: Jurnal Inovasi Dan Tren*, Vol. 1, No. 2, Pp. 228–232, Nov. 2023, Doi: 10.35870/Ljit.V1i2.2157.
- [30] A. Suharso, J. Zaelani, And D. Juardi, “Kompresi File Menggunakan Algoritma Lempel Ziv Welch (Lzw),” Vol. 17, No. 2, Pp. 372–380, 2020, [Online]. Available: <https://Journal.Unpak.Ac.Id/Index.Php/Komputasi>
- [31] A. Satyapratama And M. Yunus, “Analisis Perbandingan Algoritma Lzw Dan Huffman Pada Kompresi File Gambar Bmp Dan Png,” 2019.
- [32] M. I. Nawawi And F. Nurpandi, “Evaluation Of Deflate Algorithm In Lossless Compression Of Digital Document Formats,” 2025.
- [33] J. Adisantoso, D. Dimas Sulistio, and B. Paruhum Silalahi, “Kompresi Data Menggunakan Algoritme Huffman,” 2021.