

IMPLEMENTASI ALGORITMA ARITHMETIC CODING PADA APLIKASI COMPRESS FILE PDF

Alfira Zahwa¹, Mhd. Zulfansyuri Siambaton², Antoni³

Fakultas Teknik Universitas Islam Sumatera Utara^{1,2,3}

Email: alfirazahwa17@gmail.com

Informasi

Abstract

Volume : 2
Nomor : 10
Bulan : Oktober
Tahun : 2025
E-ISSN : 3062-9624

Di era digital, kebutuhan akan efisiensi penyimpanan dan transfer data menjadi semakin penting, khususnya untuk file berukuran besar seperti PDF. Penelitian ini membahas implementasi algoritma Arithmetic Coding pada aplikasi kompresi file PDF sebagai solusi untuk mengurangi ukuran file tanpa mengurangi kualitas konten. Arithmetic Coding merupakan metode kompresi data lossless yang bekerja dengan memetakan rangkaian simbol menjadi sebuah bilangan pecahan dalam interval $[0,1)$, kemudian dikonversi ke bentuk biner. Algoritma ini dikenal efisien karena mampu memberikan rasio kompresi yang tinggi, terutama pada data dengan tingkat redundansi yang besar. Aplikasi yang dikembangkan dirancang untuk memproses file PDF, mengekstraksi konten teks, kemudian mengompresinya menggunakan Arithmetic Coding. Hasil pengujian menunjukkan bahwa implementasi algoritma ini dapat mengurangi ukuran file PDF dengan tingkat efektivitas yang bervariasi, tergantung pada karakteristik dan kompleksitas data dalam dokumen. Dengan demikian, aplikasi ini dapat membantu pengguna dalam mengelola file PDF, khususnya pada kondisi dengan keterbatasan bandwidth atau kapasitas penyimpanan.

Keyword: Arithmetic Coding, Kompresi Data, File PDF, Aplikasi Berbasis Web.

A. PENDAHULUAN

Pemanfaatan teknologi sebagai bagian kehidupan dalam dasawarsa terakhir berkembang dengan pesat. Hal tersebut berpengaruh kepada meningkatnya kebutuhan akan media penyimpanan data. Data berupa file teks, gambar, suara, video ataupun file yang mempunyai fungsi tertentu membutuhkan lokasi sebagai wadah yang dapat dihitung dalam satu byte. Masalah dalam biaya penyimpanan, biaya dan waktu penransmisian data mengharuskan adanya suatu pemecahan. (Patanan & Saputra, 2023)

Kompresi adalah teknik untuk memperkecil ukuran data dari hasil data aslinya. Kompresi data adalah cara untuk mengurangi atau mengubah ukuran file sebelum menyimpan atau memindahkan data ke media penyimpanan. Aliran data input atau data asli di aliran data lain, atau data yang telah digunakan dan memiliki ukuran lebih kecil. Kompresi adalah proses pengurangan ukuran data atau file untuk menghasilkan repesntasi digital yang padat yang

masih mengandung atau mewakili jumlah informasi yang terkandung dalam data . Kompresi juga data dapat diartikan sebagai proses memampatkan atau mengecilkan ukuran file sebuah data, baik itu gambar, audio, ataupun video. Manfaat kompresi untuk teknologi yaitu kompresi memiliki keuntungan dalam hal penyimpanan data, Ketika kompresi data selesai, secara otomatis kita hanya membutuhkan kapasitas penyimpanan yang lebih kecil, yang tidak hanya dianggap lebih efisien daripada kompresi data, tetapi juga membantu pertukaran data lebih cepat, dan pengiriman data semakin cepat dan lebar bidang (Bandwith) dapat diperkecil. (Ambarita, 2024)

Dalam melakukan kompresi tersebut dapat dilakukan dengan menggunakan metode kompresi yang cukup populer. Yaitu Arithmetic Coding. Metode ini banyak sekali variasinya namun cukup sederhana. Arithmetic Coding dapat dikembangkan dengan menggunakan bahasa pemrograman Python yang sudah sangat populer saat ini.

Salah satu algoritma yang dapat digunakan dalam melakukan kompresi file adalah Algoritma Arithmetic Coding. Algoritma ini memberikan ide alternatif yang ada dengan melakukan proses pengkodean berupa menggantikan setiap simbol yang masuk dengan code word, sehingga dapat meningkatkan efektivitas teknik kompresi yang sudah ada pada saat ini. Algoritma ini merupakan kombinasi dari beberapa algoritma lainnya, dengan mengkombinasikan file PDF dengan algoritma Arithmetic Coding dapat diketahui pengaruh transformasi dalam pengkompresian data dan yang paling efektif dalam memampatkan berbagai jenis data.

Algoritma Arithmetic Coding merupakan teknik kompresi bersifat lossless, yang artinya menghasilkan data yang identic dengan data aslinya dan tidak berubah atau hilang pada saat proses atau dekompresi. Algoritma arithmetic Coding melakukan penggantian suatu deretan symbol input dengan sebuah bilangan floatingpoint. Semakin panjang dan semakin kompleks pesan yang dikodekan, akan semakin banyak bit yang diperlukan untuk keperluan tersebut (Gowasa & Simanjuntak, 2024).

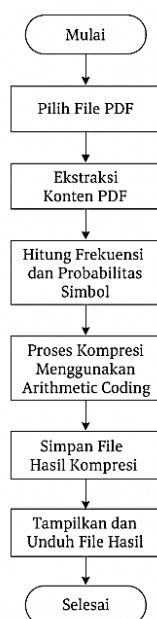
Berdasarkan uraian latar belakang di atas, rumusan masalah dalam penelitian ini adalah: (1) Bagaimana mengimplementasikan algoritma Arithmetic Coding pada aplikasi compress file PDF? dan (2) Bagaimana mengoptimalkan proses compress file besar menjadi file kecil menggunakan Arithmetic Coding tanpa merusak isi file pada aplikasi compress file PDF? Untuk menjaga fokus penelitian agar tetap sesuai dengan judul tugas skripsi, penulis memberikan batasan masalah sebagai berikut: (1) Aplikasi yang dibangun merupakan aplikasi berbasis web, (2) pembuatan aplikasi compress file PDF menggunakan bahasa pemrograman Python, (3)

aplikasi ini hanya berfungsi untuk mengubah file besar menjadi lebih kecil tanpa merusak isi file, (4) aplikasi hanya dapat memproses file dalam format PDF, dan (5) aplikasi ini hanya mengompres file PDF yang berisi teks tanpa gambar.

Tujuan dari penelitian ini adalah untuk mengimplementasikan algoritma Arithmetic Coding pada aplikasi compress file PDF guna menghasilkan tingkat akurasi yang tinggi dalam proses konversi file besar menjadi file kecil tanpa merusak isi file. Selain itu, penelitian ini juga bertujuan untuk mengoptimalkan kinerja aplikasi compress file PDF menggunakan algoritma Arithmetic Coding agar dapat memberikan hasil kompresi yang lebih efisien dan efektif.

B. METODE PENELITIAN

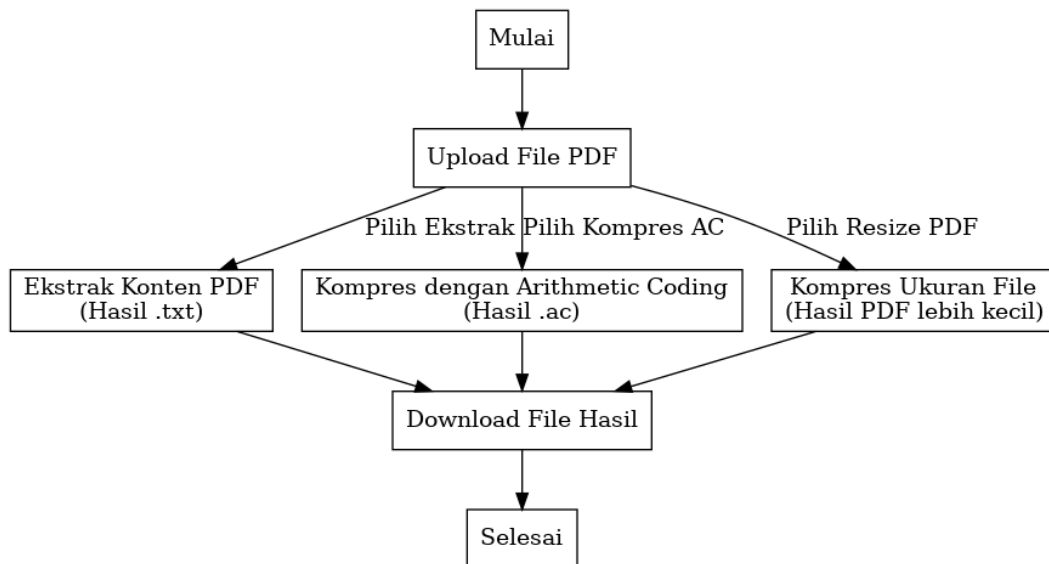
Dalam penyusunan skripsi ini, penulis memerlukan beberapa peralatan pendukung yang terdiri dari perangkat keras dan perangkat lunak untuk menunjang proses perancangan serta implementasi aplikasi. Perangkat keras yang digunakan adalah laptop dengan spesifikasi sebagai berikut: prosesor 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42GHz, kartu grafis Intel(R) Iris(R) Xe Graphics, memori 16 GB, dan penyimpanan 512 GB. Sementara itu, perangkat lunak yang digunakan meliputi sistem operasi Windows 11 64-bit, text editor Visual Studio Code, bahasa pemrograman Python, serta web browser Microsoft Edge, Google Chrome, dan Mozilla Firefox. Kombinasi perangkat keras dan perangkat lunak tersebut digunakan untuk memastikan proses pengembangan aplikasi compress file PDF dapat berjalan secara optimal dan efisien.



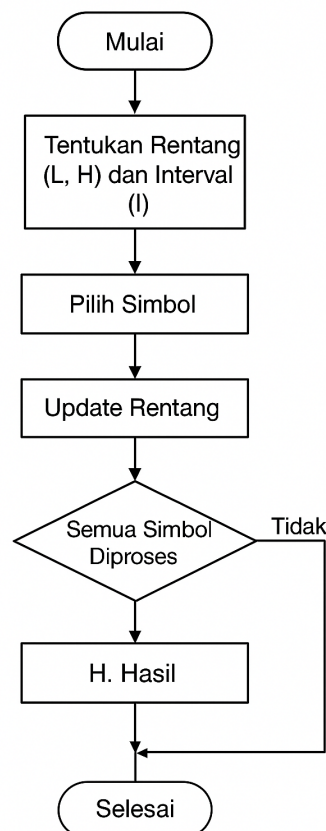
Gambar 1 Flowchart Kerja Sistem

Diagram ini menunjukkan urutan langkah-langkah dari pengguna memilih file hingga sistem menghasilkan file hasil kompresi.

Dalam melakukan penelitian ini peneliti melakukan tahapan-tahapan kegiatan dengan mengikuti rencana kegiatan yang tertuang dalam tahapan pelaksanaan penelian meliputi metode-metode pengujian analisis yang dapat dilihat pada gambar



Gambar 2 Flowchart Aplikasi



Gambar 3 Flowchart Cara Kerja Arithmetic Coding

Perancangan sistem merupakan tahap untuk merancang struktur dan alur kerja aplikasi sebelum diimplementasikan ke dalam kode program. Pada tahap ini dilakukan proses perencanaan terhadap antarmuka pengguna, rancangan arsitektur sistem, serta rancangan logika kompresi menggunakan algoritma Arithmetic Coding. Tujuan utama perancangan sistem ini adalah untuk memastikan aplikasi dapat berjalan sesuai dengan kebutuhan pengguna, yaitu melakukan kompresi file PDF secara efisien tanpa mengubah isi dokumen.

Aplikasi kompresi file PDF ini dikembangkan menggunakan pendekatan Client-Server Architecture, di mana bagian client (frontend) berfungsi sebagai antarmuka pengguna berbasis web yang dibangun menggunakan HTML, CSS, dan JavaScript, sedangkan server (backend) menggunakan Flask (Python) untuk menangani logika utama, termasuk ekstraksi teks dari PDF dan proses kompresi dengan algoritma Arithmetic Coding. Alur kerja sistem dimulai ketika pengguna membuka halaman aplikasi melalui browser, memilih file PDF, lalu mengirimkannya ke server. Server mengekstrak teks dari file menggunakan PyMuPDF, kemudian data teks dikompresi menggunakan algoritma Arithmetic Coding. Setelah proses selesai, server mengirimkan hasil kompresi kembali ke pengguna, yang dapat mengunduh file PDF terkompresi. Aplikasi ini bersifat stateless, sehingga tidak menyimpan data secara permanen di server, namun tetap menggunakan struktur data sementara untuk menyimpan nama file PDF, ukuran file sebelum dan sesudah kompresi, serta waktu proses, agar pengguna dapat melihat hasil perbandingan ukuran file.

Antarmuka pengguna aplikasi dirancang sederhana dan mudah digunakan, dengan beberapa komponen utama seperti halaman utama, tombol unggah file PDF, tombol ekstraksi konten PDF, tombol kompresi dengan Arithmetic Coding, serta tombol untuk menampilkan hasil perbandingan dan mengunduh file hasil kompresi. Tampilan hasil menampilkan nama file, ukuran awal, ukuran hasil, dan rasio kompresi, disertai notifikasi keberhasilan proses. Desain UI menggunakan warna dominan biru muda dan putih dengan tata letak responsif agar nyaman digunakan di berbagai perangkat. Proses inti sistem menggunakan algoritma Arithmetic Coding dengan tahapan meliputi penerimaan data teks, analisis frekuensi simbol, penentuan interval awal ($L=0$ dan $H=1$), penyempitan interval berdasarkan probabilitas simbol, dan menghasilkan nilai desimal akhir sebagai hasil kompresi. Alur kerja sistem divisualisasikan melalui flowchart yang menggambarkan langkah-langkah mulai dari pemilihan file PDF, ekstraksi teks, perhitungan frekuensi simbol, proses kompresi, penyimpanan hasil, hingga menampilkan rasio hasil kompresi kepada pengguna.

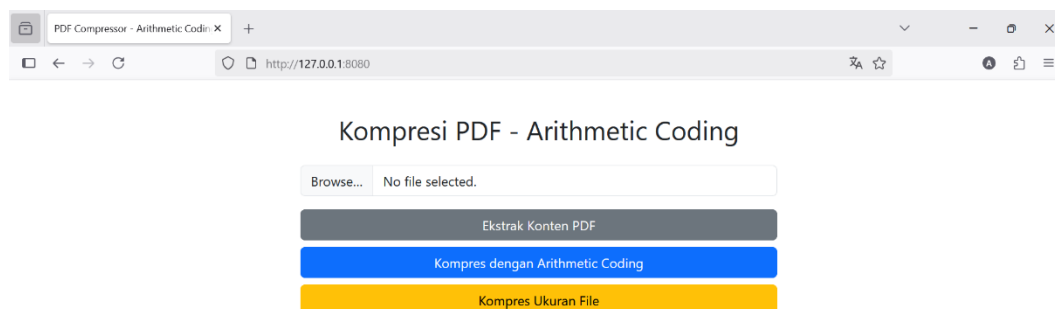
C. HASIL DAN PEMBAHASAN

Implementasi Aplikasi

Implementasi aplikasi *Compress File PDF* merupakan aplikasi yang digunakan untuk mengompresi *file* PDF agar *file* tersebut menjadi *file* yang berukuran kecil.

Tampilan Aplikasi

Setelah dilakukan perancangan dan implementasi program, berikut adalah tampilan antarmuka aplikasi kompresi PDF menggunakan algoritma *Arithmetic Coding*.

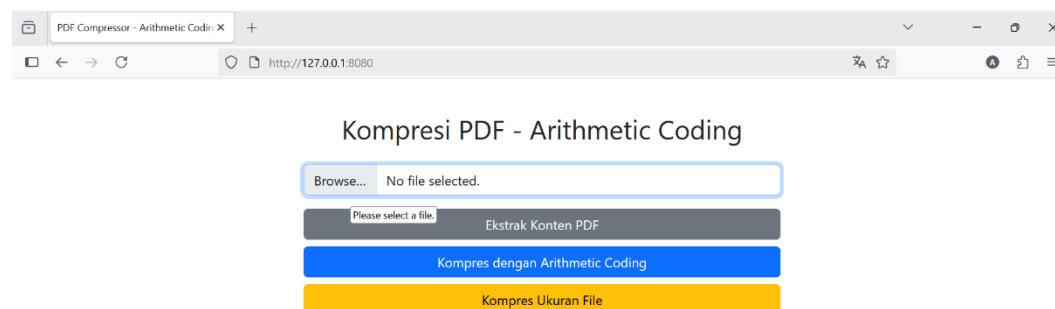


Gambar 4 Tampilan Halaman Utama Aplikasi

Pada halaman ini tersedia tiga menu utama yaitu: Ekstrak Konten PDF, Kompres dengan *Arithmetic Coding*, dan Kompres Ukuran *File*. Menu ini menjadi pintu utama pengguna untuk memilih proses yang akan dijalankan.

Proses Pemilihan File PDF

Pengguna dapat memilih *file* PDF yang akan dikompresi dengan menekan tombol Browse. Setelah *file* dipilih, aplikasi menampilkan informasi bahwa *file* telah siap untuk diproses.

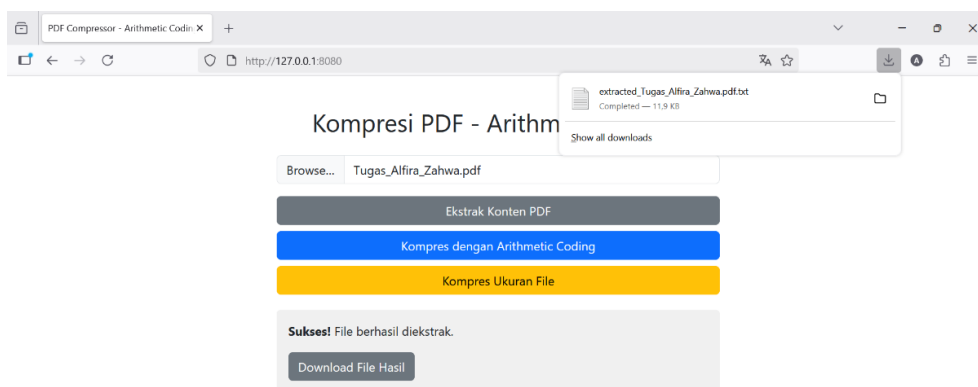


Gambar 5 Tampilan Setelah *File* Dipilih

Pada tahap ini, aplikasi akan menampilkan nama *file* serta ukuran awal *file* yang akan diproses. Langkah ini merupakan tahap awal sebelum dilakukan ekstraksi atau kompresi.

Hasil Ekstraksi Konten PDF

Gambar ini menampilkan hasil dari menu Ekstrak Konten PDF. Aplikasi mengekstrak teks dari *file* PDF dan menyimpannya ke dalam *file* berformat .txt.

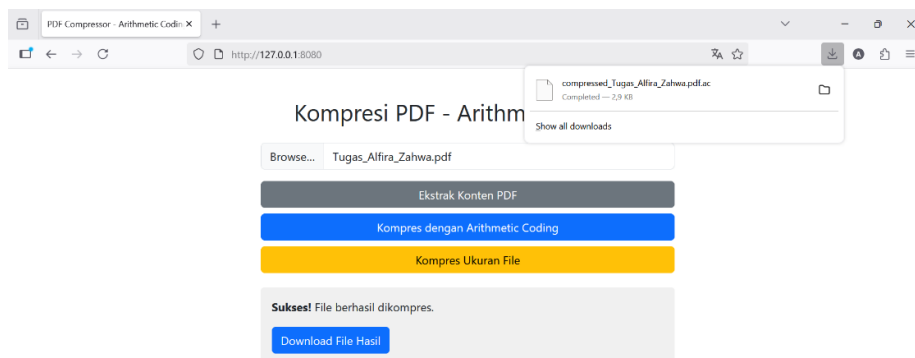


Gambar 6 Menu Ekstrak Konten PDF

Pada tampilan ditunjukkan pesan *“Sukses! File berhasil diekstrak”* serta tombol Download File Hasil. Fungsi ini berguna jika pengguna hanya memerlukan isi *teks* dari dokumen PDF.

Hasil Ekstraksi Kompresi Dengan Arithmetic Coding

Gambar ini menunjukkan hasil dari proses Kompresi dengan *Arithmetic Coding*. File PDF berhasil dikompresi menjadi *file* dengan ekstensi *.ac*.

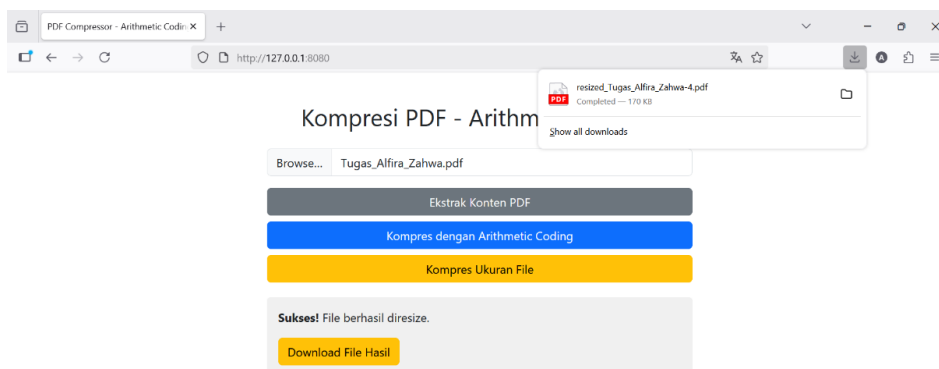


Gambar 7 Menu Kompres Dengan Arithmetic Coding

Pesan *“Sukses! File berhasil dikompres”* muncul bersama tombol Download File Hasil. Menu ini menjadi bukti implementasi algoritma *Arithmetic Coding* pada aplikasi.

Hasil Kompresi Ukuran File PDF

Gambar ini memperlihatkan hasil dari menu Kompres Ukuran File.



Gambar 8 Menu Kompres Ukuran File

Aplikasi menghasilkan *file* PDF baru dengan ukuran lebih kecil, ditandai dengan pesan sukses dan tombol unduh. Fitur ini memudahkan pengguna untuk menyimpan PDF dalam ukuran yang lebih efisien.

Perbandingan Ukuran File Asli dan Hasil Kompresi

Setelah dilakukan kompresi menggunakan aplikasi, *file* PDF yang dipilih menghasilkan file baru dengan ukuran lebih kecil dibandingkan file asli.

Tabel 1 Perbandingan Ukuran *File* Asli dan Hasil Kompresi

NO	NAMA FILE ASLI	UKURAN FILE SEBELUM DI KOMPRESI	NAMA FILE HASIL KOMPRESI	UKURAN SESUDAH DI KOMPRESI	PENGURANGAN	Jumlah String (Karakter)
1.	1.CV_F.pdf	1,46 MB	resized_1.CV_F.pdf	154 KB	89,45 %	± 46.000
2.	LAPORAN_AKHIR_STUPEN_kampus.pdf	1,17 MB	resized_LAPORAN_AKHIR_STUPEN_kampus.pdf	1,15 MB	1,70 %	± 18.500
3.	Tugas_Rangkuman_Teknologi_Opensource.pdf	1,95 MB	resized_Tugas_Rangkuman_Teknologi_Opensource.pdf	107 KB	94,51 %	± 65.000
4.	Tugas_Alfira_Zahwa.pdf	174 KB	resized_Tugas_Alfira_Zahwa.pdf	170 KB	2.30 %	± 9.500
5.	judul1fira.pdf	423 KB	resized_judul1fira.pdf	400 KB	5.43 %	± 21.300
6.	Jurnal_TA.pdf	299 KB	resized_Jurnal_TA.pdf	289 KB	3.34 %	± 15.700

Tabel 1 menunjukkan hasil perbandingan ukuran file PDF sebelum dan sesudah dilakukan proses kompresi menggunakan algoritma *Arithmetic Coding* yang diimplementasikan pada aplikasi. Dari tabel terlihat bahwa setiap file mengalami penurunan ukuran, meskipun dengan persentase yang berbeda-beda.

1.CV_F.pdf

Ukuran awal file adalah 1,46 MB, dan setelah dikompresi menjadi 154 KB. Terjadi penurunan ukuran sebesar 89,45%. Hasil ini menunjukkan bahwa file *1.CV_F.pdf* memiliki dominasi teks dengan tingkat pengulangan karakter yang tinggi. Hal tersebut membuat algoritma *Arithmetic Coding* mampu melakukan kompresi secara maksimal dengan

mengurangi redundansi data. Penurunan yang signifikan ini menandakan bahwa isi dokumen efisien untuk dikompresi tanpa mengubah konten asli file.

LAPORAN_AKHIR_STUPEN_kampus.pdf

Ukuran awal file adalah 1,17 MB, sedangkan ukuran setelah kompresi menjadi 1,15 MB. Penurunan yang terjadi sebesar 1,70%. Hasil ini menunjukkan bahwa file ini memiliki banyak elemen non-teks seperti gambar, grafik, atau tabel berwarna yang sebelumnya telah mengalami kompresi lossy (misalnya JPEG). Karena itu, algoritma Arithmetic Coding tidak dapat mengurangi ukuran file secara signifikan. Meskipun begitu, hasil kompresi tetap menunjukkan bahwa sistem bekerja dengan baik dan tidak merusak struktur dokumen.

Tugas_Rangkuman_Teknologi_Opensource.pdf

Ukuran awal file adalah 1,95 MB, dan setelah dikompresi menjadi 107 KB. Penurunan ukuran mencapai 94,51%, yang merupakan hasil kompresi paling besar di antara semua file yang diuji. Hal ini disebabkan isi file sebagian besar berupa teks polos tanpa gambar, sehingga algoritma Arithmetic Coding mampu menekan redundansi simbol secara optimal. Kompresi ini sangat efisien dan membuktikan bahwa algoritma dapat menghasilkan file berukuran jauh lebih kecil tanpa kehilangan data.

Tugas_Alfira_Zahwa.pdf

Ukuran awal file adalah 174 KB, dan setelah dikompresi menjadi 170 KB. Terjadi penurunan sebesar 2,30%. Hal ini menunjukkan bahwa dokumen masih memiliki redundansi data yang dapat dikompresi, meskipun pengurangannya tidak terlalu signifikan.

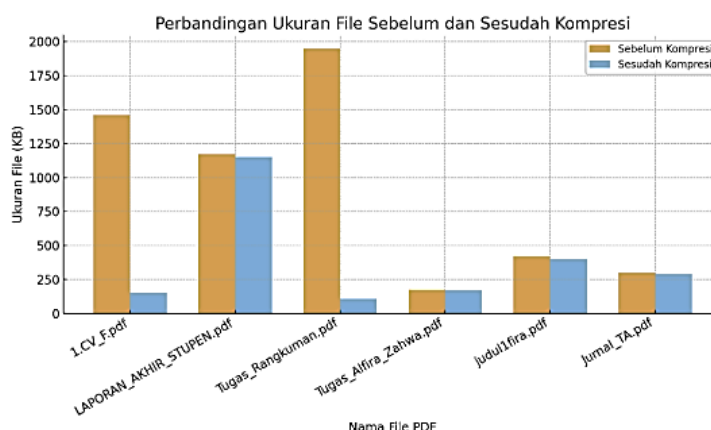
judul1fira.pdf

Ukuran awal file adalah 423 KB, dan setelah kompresi menjadi 400 KB, dengan pengurangan 5,43%. Ini merupakan persentase penurunan terbesar di antara sampel yang diuji, yang berarti file ini memiliki lebih banyak data redundan yang berhasil dikompresi oleh algoritma.

Jurnal_TA.pdf

Ukuran awal file adalah 299 KB, sedangkan ukuran setelah kompresi adalah 289 KB. Penurunan yang terjadi adalah sebesar 3,34%. Hasil ini menunjukkan bahwa meskipun file relatif kecil, kompresi tetap dapat mengurangi ukuran file.

Secara keseluruhan, hasil uji coba membuktikan bahwa implementasi algoritma *Arithmetic Coding* dapat mengurangi ukuran file PDF, meskipun tingkat efektivitasnya dipengaruhi oleh karakteristik isi dokumen. File dengan banyak teks cenderung lebih mudah dikompresi dibanding file yang didominasi oleh gambar atau elemen yang sudah menggunakan kompresi internal.



Gambar 9 Diagram Batang Hasil Perbandingan Ukuran File sebelum dan sesudah kompresi menggunakan algoritma Arithmetic Coding.

D. KESIMPULAN

Algoritma *Arithmetic Coding* berhasil diterapkan pada aplikasi kompresi *file* PDF dan mampu mengurangi ukuran *file* secara signifikan tanpa mengurangi kualitas konten di dalam *file* PDF tersebut. Aplikasi yang dibangun mampu membaca konten teks dari *file* PDF, melakukan proses encoding menggunakan algoritma *Arithmetic Coding*.

E. DAFTAR PUSTAKA

- Abdurahman, H., & Riswaya, A. R. (2014). APLIKASI.
- Ambarita, C. N. (2024). Analisis Perbandingan Algoritma Run Length Encoding Dan Algoritma Five Modulus Dalam Kompresi File PDF. JURNAL INFORMATIKA, 01(03).
- Anendya, A. (2023, July 22). Mengenal Apa itu Aplikasi, Fungsi, dan Jenis-Jenisnya. <https://www.dewaweb.com/blog/apa-itu-aplikasi/>
- Gowasa, F. S., & Simanjuntak, A. (2024). Jurnal Interaksi Komputer. 1.
- Patanan, J., & Saputra, R. A. (2023). APLIKASI KOMPRESI FILE AUDIO MENGGUNAKAN ALGORITMA DEFLATE.
- Qifli Ilhamdi, J., Julkarnain, M., & Yuliadi, Y. (2024). SISTEM INFORMASI ADMINISTRASI UKM RACANA OLAT MARAS – AI RENUNG UNIVERSITAS TEKNOLOGI SUMBAWA BERBASIS WEBSITE. JATI (Jurnal Mahasiswa Teknik Informatika), 8(3), 4309–4315. <https://doi.org/10.36040/jati.v8i3.9905>
- Sianturi, F. A. (2018). Kompresi File Citra Digital Dengan Arithmetic Coding. 03.
- Syahrudin, A. N., & Kurniawan, T. (2018). INPUT DAN OUTPUT PADA BAHASA PEMROGRAMAN

PYTHON.

Syifaudin, E. (2022, October 26). Jenis Website di Internet Berdasarkan Fungsi, Platform dan Sifat. <https://www.exabytes.co.id/blog/jenis-website-di-internet/>

Tari, R., Nasution, S. D., & Zebua, T. (2021). Penerapan Algoritma Arithmetic Coding Untuk Mengompresi Record Database Pada Aplikasi Ensiklopedia Flora Berbasis Android. 2(7).