Jurnal Sains Dan Teknologi Informasi
Vol 3, No 2, Maret 2024, Hal 53-60
ISSN 2809-610X (Media Online)
DOI 10.47065/jussi.v3i2.4884
Website https://ejurnal.seminar-id.com/index.php/jussi

Implementasi Algoritma Arithmetic Coding Pada AplikasiKompresi File PDF

Egi Prades

Program Studi Teknik Informatika, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Budi Darma,
Jalan Sisingamanganraja No. 338, Medan, Sumatera Utara, Indonesia
Email: egiprades7@gmail.com

Abstrak-Saat ini pengguna aplikasi sebagai informasi semakin umum digunakan. Tetapi ada masalah yang sering dijumpai , yaitu kebutuhan data yang besar sehingga ruang penyimpanan yang dibutuhkan semakin sedikit, selain itu dengan sedikitnya ruang penyimpanan juga mempengaruhi kecepatan dalam pengirimam data salah satu data yang sering digunakan adalah file format PDF. Adapun solusi dalam masalah ini yaitu dengan melaukukan kompresi terhadap file PDF, kompresi yaitu sebbuah proses mengubah sekumpulan data menjadi suatu bentuk code untuk menghemat kebutuhan penyimpanan data sehinga data yang digunakan tidak terlalu besar. Algoritma Arithemtic Coding adalah salah satu algotitma kompresi data yang dapat digunakan, dengan mengatahui hasil proses kompresi rasio dan *redudancy* nya. Dalam penerapan algoritma *Aritmetic Coding* pada penelitian ini, dapat memberikan hasil kompresi yang awalnya mempunyai ukuran besar dapat terkompres dengan sangat baik pada file PDF dan hasil kompresi menunjukan hasil *ratio compression* 2,15 sedangkan *Compresion ratio* 46% dan *Redudancy* 54% akan menguntungkan dalam melakukan pengiriman dan pemindahan *file* PDF akan semakin mudah.

Kata Kunci: Kompresi; File PDF; Algoritma Arithmetic Coding

Abstract-Currently, application users use information more and more commonly. However, there is a problem that is often encountered, namely the need for large data so that less storage space is needed. Apart from that, the small amount of storage space also affects the speed of sending data. One of the data that is often used is PDF format files. The solution to this problem is to compress the PDF file. Compression is a process of changing a collection of data into a form of code to save on data storage requirements so that the data used is not too large. The Arithemtic Coding algorithm is a data compression algorithm that can be used, by knowing the results of the compression ratio and redundancy process. In applying the Arithmetic Coding algorithm in this research, it can provide compression results which initially have a large size which can be compressed very well in PDF files and the compression results show a compression ratio of 2.15 while the Compression ratio is 46% and Redundancy 54% will be beneficial in sending and moving PDF files will be even easier.

Keywords: Compression; PDF Files; Arithmetic Coding Algorithm

1. PENDAHULUAN

Dengan kemajuan teknologi yang semakin pesat pada saat ini perlu meningkatkan kecepatan pengiriman informasi, karena semakin maraknya kebutuhan akan informasi dari setiap orang. Segingga diperlukan proses yang cepat, tentunya semakin banyak pula data-data atau *file-file* yang ingin disimpan dan ingin menyimpan *file* tersebut, setiap *file* memiliki ukuran masing-masing tergantungdikirim. *File* dikomputer pada umumnya disimpan pada suatu folder tertentu dimana pengguna jenis *file* itu sendiri seperti *file* Gambar, *file* audio, *file* dokumen dan lainnya. Adapun *File* yang sering digunakan salah satunya adalah *file Portable Document Format* (PDF).

Dalam penggunaan *file* PDF juga memiliki kekurangan bagi pengguna sebab ukurannya terkadang besar sehingga membuat kapasitas memori penuh. Oleh karena itu diperlukan kompresi terlebih dahulu supaya format *file* tersebut menjadi ukuran yang lebih kecil. Apabila ukuran *file* dapat dikompres menjadi kecil dari ukuran aslinya, maka secara otomatis penyimpanan memori dapat menyimpan data lebih banyak.

Kompresi *file* PDF mengecilkan/memapatkan ukuran data sehingga data yang dikompres dalam bentuk PDF menjadi lebih kecil dari ukuran aslinya. Dengan melakukan kompresi, data yang sebelumnya besar ukurannya akan diperkecil sehingga dapat menghemat kapasitas penyimpanan danwaktu untuk pertukaran/transimi data.

Ada beberapa faktor yang harus diperhatikan dalam mengkompresi data yaitu, *Ratio Of Compression*, *Compression Ratio*, Redudancy. Selain itu aplikasi yang digunakan dalam melakukan kompresi bermacam-macam terkadang aplikasi yang digunakan untuk melakukan kompresi tidak semuanya berhasil, ukuran file yang dikompres tidak berubah menjadi lebih kecil melainkan tetap diukuran normal hal ini disebabkan ketidak sempurnaan aplikasi yang digunakan dan algoritma kompresi yang diterapkan haruslah relevan. Dalam melakukan kompresi terdapat banyak sekali Algoritma yang bisa digunakan salah satunya adalah algoritma *Arithmetic Coding*.

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 codeword, 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.

Berdasarkan penelitian terdahulu bahwa algoritma *Arithmetic Coding* pengkompresian data video dan audio pada proses kompresinya menghasilkan hasil kompresian yang sama, kecepatan dalam proses kompresi tidak hanya diukur dengan jumlah *file* audio dan videonya saja melainkan ukuran file yang akan dikompres selain itu waktu yang diperlukan dalam mengkompres *file* MP4 lebih besar dibandikangkan dengan waktu kompres*file* MP3, algoritma *Arithmetic Coding* tidak

Jurnal Sains Dan Teknologi Informasi
Vol 3, No 2, Maret 2024, Hal 53-60
ISSN 2809-610X (Media Online)
DOI 10.47065/jussi.v3i2.4884
Website https://ejurnal.seminar-id.com/index.php/jussi

optimal digunakan pada kompres *file* audio dan video [1].

Algoritma *arithmetic coding* merupakan teknik kompresi bersifat *lossless*, yang artinya menghasilkan data yang identic dengan data aslinya dan tidak berubah atau hilang pada saat prosesatau dekompresi [2].

2. METODOLOGI PENELITIAN

2.1 Kompresi File PDF

Kompresi data adalah proses mengubah satu aliran data input (sumber atau data mentah asli) menjadi aliran data lain (menghasilkan aliran bit atau aliran terkompresi) dari ukuran yang lebih kecil. Kompresi File PDF adalah metode yang ada dikomputer yang bertujuan untuk mengompresi data sehingga membutuhkan ruang penyimpanan yang lebih sedikit [3].

Tujuan kompresi data adalah untuk merepresentasikan data digital dengan sejumlah kecil bit, sambil mempertahankan persyaratan pembentukan ulang minimal dari data asli. Data digital ini dapatberupa file teks, file gambar, audio, dan kombinasi ketiganya, seperti file video.

2.2 Algoritma Arithmetic Coding

Secara umum, algoritma kompresi data menerapkan teknik encoding dengan mengganti satu atau lebih simbol dengan kode yang sama, tidak seperti pengkodean *Arithmetic Coding*, yang mengkodekan semua aliran input menjadi satu kode teks dengan angka floating point dengan interval (0,1) atau angka yang lebih besar atau sama dengan 0 dan lebih kecil dari 1($0 \leq x < 1$).

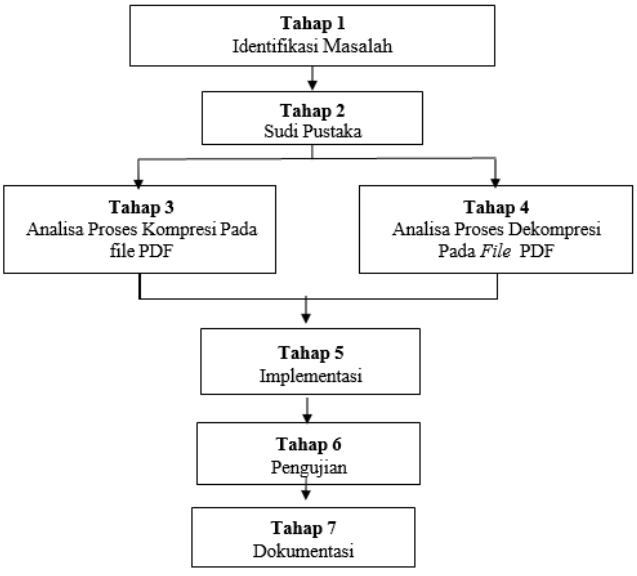
Aritmetic Coding memiliki sejarah yang penting karena pada saat ini algoritma ini telah menggantikan algoritma *Huffman* selama 25 tahun, *Arithmetic Coding* mengungguli *Huffman Coding* terutama jika diterapkan pada sekumpulan alfabet yang relatif kecil, Awalnya, *Arithmetic Coding* diperkenalkan oleh Shannon Fano dan Elias, kemudian dikembangkan secara luas oleh Pasco (1976), Rissanene (1976, 198) dan Langdon (198), yaitu sebuah ide alternatif, masing-masing memasukkan symbol dengan *Codeword*, yaitu dengan mengkodekan seluruh input dengan sebuah floating point yang merupakan output dari kompresi. Untuk melakukan kompresi dan dekompresi, *Arithmetic Coding* memerlukan beberapa langkah untuk melakukan kompresi, yaitu pertama menentukan nilai probabilitas kemunculan, menentukan nilai range setiap simbol, kemudian melakukan encoding dan decoding [4].

Adapun langkah *Arithmetic Coding* pada penelitian ini yaitu :

1. Mengambil sampel sebanyak 20 Sampel dari *Software HXD*
2. Melakukan pendataan simbol *Arithmetic Coding*
3. Menentukan nilai probabilitas yang muncul dari setiap simbol
4. Menentukan nilai range
5. Dilanjutkan kompresi menggunakan *Arithmetic Coding*
6. Melakukan rescaling dari hasil kompresi

2.3 Tahapan Penelitian

Tahapan penelitian menggambarkan kerangka penelitian atau yang biasa disebut dengan tahapan penelitian. Tahapan ini terdiri dari sejumlah langkah sistematis yang saling berhubungan. Tahapan penelitian ini diperlukan untuk memudahkan pelaksanaan penelitian. Adapun tahapan penelitian yang dilakukan yaitu:



Gambar 1. Tahapan Penelitian

Jurnal Sains Dan Teknologi Informasi
Vol 3, No 2, Maret 2024, Hal 53-60
ISSN 2809-610X (Media Online)
DOI 10.47065/jussi.v3i2.4884
Website https://ejurnal.seminar-id.com/index.php/jussi

Berdasarkan diagram tahapan penelitian pada gambar 1, maka dapat diuraikan pembahasan dari masing-masing tahapannya sebagai berikut :

a. Tahap Identifikasi Masalah
   Langkah ini adalah cara penulis memprediksi, memperkirakan dan menggambarkan apa yang menjadi sumber masalah dalam kapasitas penyimpanan yang memadai sehingga terjadinya kelambatan pada saat transfer file Pdf bahkan sampai tidak dapat lagi disimpan di ruang penyimpanan.
b. Tahap Studi Pustaka
   Pada proses dapat dilakukan pemahaman terhadap objek yang akan diteliti dengan memahami berbagai sumber referensi seperti buku, jurnal, maupun sumber bacaan lainnya yang dapat digunakan.
c. Tahap Analisa Proses Kompresi Pada File Pdf
   Tahapan analisa merupakan tahapan untuk mengetahui proses pengkompresian file Pdf menggunakan algoritma Arithmetic Coding yang bertujuan untuk memperkecil ukuran data, penerapan algoritma Arithmetic Codimg ditujukan untuk menghitung probabilitas dari setiap frekuensi yang ada.
d. Tahap Analisa Proses Dekompresi Pada File Pdf
   Proses pengembalian file Pdf yang telah dikompresi menjadi kebentuk file asli sebelum dikompresi menggunakan algoritma Arithmetic Coding.
e. Tahap Implementasi
   Merupakan tahapan yang dapat dilakukan pada proses pembangunan perangkat lunak/coding yang bertujuan untuk mengetahui apakah sistem dapat berjalan dengan baik dan sesuai kebutuhan atau masih diperlukannya perbaikan pada sistem tersebut.
f. Tahap Pengujian
   Tahap ini dilakukan untuk menguji hasil kompresi file Pdf . Proses ini bertujuan untuk mengetahui hasil akhir dari penelitian yang dilakukan
g. Tahap Dokumentasi
   Tahap dokumentasi merupakan tahap akhir dari pelaksanaan penelitian yang dibuat dalam bentuk laporan. Tahap ini akan menjelaskan aplikasi yang telah dibuat agar memudahkan pembaca yang ingin mengembangkan aplikasi lebihlanjut.

3. HASIL DAN PEMBAHASAN

3.1 Analisa

Analisis penelitian ini adalah semakin besar kebutuhan manusia maka semakin banyak pula data yang dapat dikumpulkan oleh manusia. Kecenderungan manusia untuk mengumpulkan data telah menyebabkan penggunaan penyimpanan data yang semakin besar. Semakin besar media penyimpanan yang dibutuhkan untuk menyimpan data, semakin banyak pula kompresi data yang diperlukan Kompresi dataadalah proses pengurangan ukuran data untuk membuat representasi digital padat yang masih dapat mewakili sejumlah besar informasi tentang data. Meskipun banyak algoritma telah dikembangkan untuk kompresi data, masih belum ada algoritma yang baik untuk kompresi data. Dalam penelitian ini, penulis menggabungkan algoritma enkripsi aritmatika dalam proses kompresi *file* PDF.

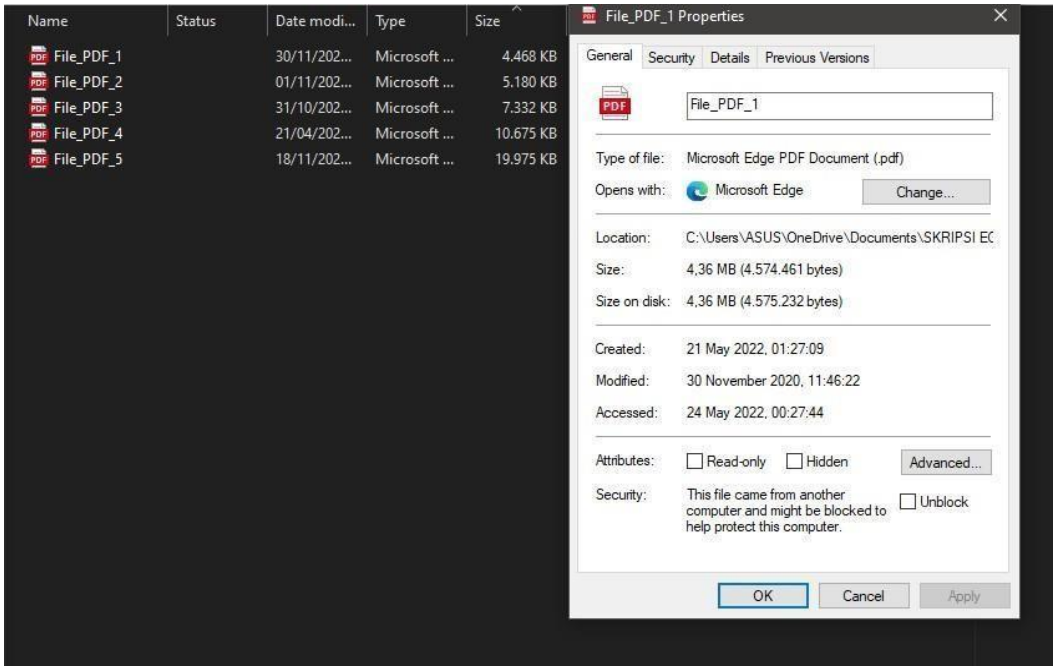
Analisis sistem merupakan langkah awal dalam penelitian yang mengeksplorasi masalah- masalah yang terlibat dalam pembuatan suatu sistem dan menggambarkan proses-proses yang ada dalam sistem untuk menghasilkan suatu keluaran yang memenuhi kebutuhan pengguna.

Pada tahap ini akan menjelaskan proses kompresi *file* PDF dimana kompresi ini akan dilakukan dengan menggunakan algoritma *Arithmetic Coding* yang merupakan sebuah algoritma yang mengubah data menjadi bentuk data menggunakanbeberapa bit, proses kerja algoritma *Arithmetic coding* yaitu dengan menggantikan suatu string simbol input dengan angka floating point. Semakin panjang dan semakin kompleks pesan yang dicodekan, semakin banyak bit yang dibutuhkan. Keluaran dari *Arithmetic Coding* ini adalah suatu bilangan yang kurang dari 1 dan lebih besar dari atau sama dengan 0, yang dapat didecodekan untuk menghasilkan deretan simbol yang digunakan untuk menghasilkan bilangan tersebut.

3.2 Penerapan Algoritma Arithmetic Coding

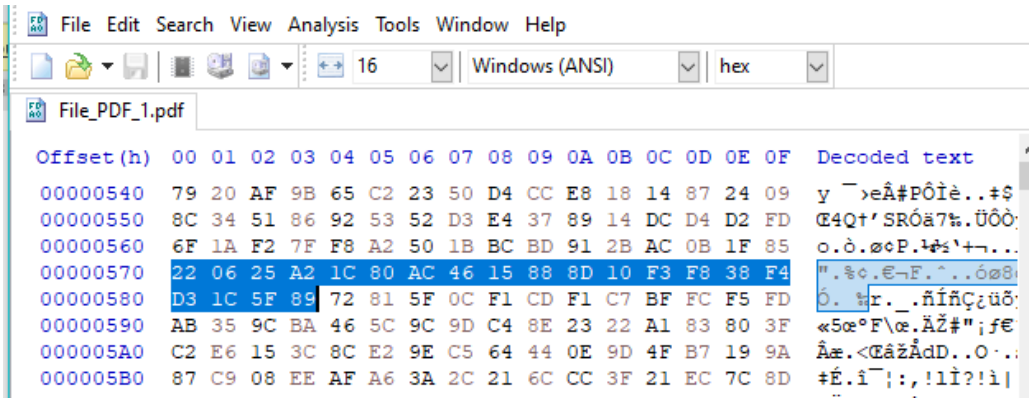
Tahap ini akan melakukan kompresi *file* PDF menerapkan algoritma *Arithmetic Coding*. Sebelum *file* dikompres peneliti terlebih dahulu mencari *file* PDF yang akan dikompres. Berikut adalah contoh *file* PDF yang akan dikompres.

Jurnal Sains Dan Teknologi Informasi
Vol 3, No 2, Maret 2024, Hal 53-60
ISSN 2809-610X (Media Online)
DOI 10.47065/jussi.v3i2.4884
Website https://ejurnal.seminar-id.com/index.php/jussi



Gambar 2. Sampel *File* PDF

Berdasarkan gambar diatas, langkah yang harus dilakukan yaitu mencari nilai *hexadecimal* pada *file* PDF dengan menggunakan aplikasi *HXD.*



Gambar 3. Nilai *Hexadecimal* Sampel *File* PDF

Berdasarkan gambar nilai *Hexadecimal* diatas, penulis telah mengambil 20 sampel nilai *Hexadecimal* untuk proses kompresi *file PDF*.

Tabel 1. Tampilan Nilai Bilangan *Hexadecimal*

22	06	25	A2	1C	80	AC	46	15	88
8D	10	F3	F8	38	F4	D3	1C	5F	89

Dari tabel diatas menghasilkan nilai *hexadecimal* dan akan dilakukan pendataan diurutkan berdasarkan frekuensi, nilai yang paling sering muncul akan ditempatkan pada urutan pertama.

Tabel 2. Pendataan Simbol *Arithmetic Coding*

NO	*Hexadecimal*	Frekuensi
1	22	1
2	06	1
3	25	1
4	A2	1
5	1C	2
6	80	1
7	AC	1
8	46	1
9	15	1

Jurnal Sains Dan Teknologi Informasi
Vol 3, No 2, Maret 2024, Hal 53-60
ISSN 2809-610X (Media Online)
DOI 10.47065/jussi.v3i2.4884
Website https://ejurnal.seminar-id.com/index.php/jussi

10	88	1
11	8D	1
12	10	1
13	F3	1
14	F8	1
15	38	1
16	F4	1
17	D3	1
18	5F	1
19	89	1

Kemudian, untuk menghitung probabilitas kemunculan setiap symbol, nilai frekuensi diperoleh dengan membagi nilai frekuensi setiap nilai heksadesimal dalam tabel dengan total 20 nilaiheksadesimal.

Tabel 3. Probabilitas Kemunculan Dari Setiap Simbol

No	*Hexadecimal*	Frekuensi	Probabilitas
1	22	1	1/20=0,05
2	06	1	1/20=0,05
3	25	1	1/20=0,05
6	80	1	1/20=0,05
7	AC	1	1/20=0,05
8	46	1	1/20=0,05
9	15	1	1/20=0,05
10	88	1	1/20=0,05
11	8D	1	1/20=0,05
12	10	1	1/20=0,05
13	F3	1	1/20=0,05
14	F8	1	1/20=0,05
15	38	1	1/20=0,05
16	F4	1	1/20 =0,05
17	D3	1	1/20=0,05
18	5F	1	1/20=0,05
19	89	1	1/20=0,05

Langkah selanjutnya yaitu menghitung titik yang paling rendah (*low*) adalah 0,0 dan menghitung titik yang paling tinggi (*high*) adalah 1,0 dengan cara pada hexsa pertama dikurangnilai probabilitas.

Tabel 4. Jangkauan Range

No	*Hexadecimal*	Frekuensi	Probabilitas	Jangkauan *Range*	
				low	*High*
1	22	1	1/20 = 0,05	0,95	1,0
2	06	1	1/20=0,05	0,9	0,95
3	25	1	1/20=0,05	0,85	0,9
4	A2	1	1/20=0,05	0,8	0,85
5	1C	2	2/20 =0,1	0,7	0,8
6	80	1	1/20=0,05	0,65	0,7
7	AC	1	1/20=0,05	0,6	0,65
8	46	1	1/20=0,05	0,55	0,6
9	15	1	1/20=0,05	0,5	0,55
10	88	1	1/20=0,05	0,45	0,5
11	8D	1	1/20=0,05	0,4	0,45
12	10	1	1/20=0,05	0,35	0,4
13	F3	1	1/20=0,05	0,3	0,35
14	F8	1	1/20=0,05	0,25	0,3
15	38	1	1/20=0,05	0,2	0,25
16	F4	1	1/20 =0,05	0,15	0,2
17	D3	1	1/20=0,05	0,1	0,15
18	5F	1	1/20=0,05	0,05	0,1
19	89	1	1/20=0,05	0,0	0,05

Langkah selanjutnya yaitu melakukan perhitungan nilai *high* dan *low* sesuai setiap simbolnya dengan rumus sebagai berikut:

Jurnal Sains Dan Teknologi Informasi
Vol 3, No 2, Maret 2024, Hal 53-60
ISSN 2809-610X (Media Online)
DOI 10.47065/jussi.v3i2.4884
Website https://ejurnal.seminar-id.com/index.php/jussi

Set low = 0.0 Set high = 1.0
While (Simbol Input masih ada) doAmbil symbol input
Coderange = high-low
High = *low – Coderange * high _Range (Symbol)*Low = *low – Coderange * Low _Range (Symbol) End While*

Tabel 5. Kompresi *Arithmetic Coding*

NO	*Hexadecimal*	*LOW/HIGH*		HASIL
1	22	*Low*	0,0 + (1,0 - 0,0)*0,95	0,95
		High	0,0 + (1,0 - 0,0) *1,0	1,0
2	6	*Low*	0,95 + (1,0 – 0,95) * 0,9	0,995
		High	0,95 + (1,0 – 0,95) * 0,95	0,9975
3	25	*Low*	0,0 + (1,0 – 0,0) *0,85	0,85
		High	0,0 + (1,0 – 0,0) * 0,9	0,9
4	A2	*Low*	0,85 + (0,9 – 0,85) *0,8	0,89
		High	0,85 + (0,9 – 0,85) *0,85	0,8925
5	1C	*Low*	0,0 + (1,0 – 0,0)*0,7	0,7
		High	0,0 + (1,0 – 0,0) *0,8	0,8
6	1C	*Low*	0,7 +(0,8 – 0,7) *0,7	0,77
		High	0,7 + (0,8 - 0,7) *0,8	0,78
7	80	*Low*	0,0 + (1,0 – 0,0) * 0,65	0,65
		High	0,0 + (1,0 – 0,0) * 0, 7	0,7
8	AC	*Low*	0,65 + (0,7 – 0,65) *0,6	0,68
		High	0,65 + (0,7-0,65) *0,65	0,6825
9	46	*Low*	0,0 + (1,0 – 0,0) *0,55	0,55
		High	0,0 + (1,0 – 0,0) *0,6	0,6
10	15	*Low*	0,55 + (0,6 – 0,55) * 0,5	0,575
		High	0,55 + (0,6 – 0,55) * 0,55	0,5775
..	....	...	....	....
20	89	*Low*	0,05 + (0,1 – 0,05) * 0,0	0,05

Representasi dari nilai *hexsa* " 22 06 25 A2 1C 80 AC 46 15 88 8D 10 F3 F8 38 F4 D3 1C 5F 89" yang telah dimapatkan dan diambil nilai *biner* yang berada diantara (0,995 - 0,9975) , (0,89 - 0,8925), (0,77 - 0,78), (0,68 - 0,6825), (0,575 - 0,5775), (0,47 - 0,4725), (0,3625 - 0,3675), (0,26 - 0,2625), (0,155- 0,1575), (0,05 – 0,0525). Nilai biner didapatkan dengan cara sebagai berikut.

a. Biner diantara (0,0995-0,9975)

Tabel 6. Rescaling Nilai Hexsa 22 dan 06

Hexsa	Low	High					
	0,995	**0,9975**					
	1,0	1,0	1,0	1,0	1,0	1,0	1,0
	X	0,5	0,75	0,875	0,9375	0,96875	0,98438
Hexsa	Low	High					
	0,995	**0,9975**					
06	**X > 0,995**	False	False	False	False	False	False
	X < 0,9975	True	True	True	True	True	True
	Nilai Biner	1	1	1	1	1	1
	1,0	1,0	1,0	0,0025			
	X	0,9921	0,996				
		9	09				
22	0,0	0,9843	0,992	0,0028			
		8	19	15			
06	**X > 0,995**	False	True				
	X < 0,9975	True	True				
	Nilai Biner	1	1				

Hasil Biner = 11111111

Berdasarkan Algoritma *Arithmetic Coding* diatas maka menghasilkan *string* bit **"11111111111001110011010111011001001110111100010101110101000011001010000 1101"** dengan total bit berjumlah 75 bit, selanjutnya membagi bit menjadi 8 bit setiap kelompoknya , dengan menyisakan 3 bit yang dengan kata lain 75 Mod 8 = 3 dengan menyatakan hasil tersebut dengan "n" dan menambahkan *padding* dan *flagging*.

Padding dengan rumus:

7 – n + "1"
7 – 3 + "1" = 4 = 00001

Jurnal Sains Dan Teknologi Informasi
Vol 3, No 2, Maret 2024, Hal 53-60
ISSN 2809-610X (Media Online)
DOI 10.47065/jussi.v3i2.4884
Website https://ejurnal.seminar-id.com/index.php/jussi

Flagging dengan rumus:

9 – n

9 – 4 = 6 = 00000110

Lanjut penggabungan *padding* dan *flagging* sehingga menfhasilkan *string* bit dengan total 88 bit.

"11111111111001110011010111011001001110111100010101110101000011001010000110100000 0100000110"

Selanjutnya yaitu menghitung *ratio of compresion*, *compression ratio*, dan *redundancy* untuk mengetahui kinerja kompresi data yang sebelumnya dengan jumlah bit 190 bit dan setelah dikompres menjadi 88 bit.

a. *Ratio Of Compression (RC)*

$$RC = \frac{Ukuran\ Data\ Sebelum\ Dikompresi\ Ukuran}{Data\ Setelah\ Dikompresi}$$

$$RC = \frac{190}{88} = 2{,}15$$

b. *Compresion Ratio (CR)*

$$CR = \frac{Ukuran\ Data\ \ Setelah\ Dikompresi\ Ukuran\ Data}{Ukuran\ Data\ Sebelum\ Dikompresi}\ X\ 100\%$$

$$CR = \frac{88}{190}\ X\ 100\% = 46\%$$

c. *Redudancy (Rd) RD* = 100% - CR *RD* = 100% - 46%*RD* = 54%

Dari perhitungan diatas dapat disimpulkan bahwa persentase hasil kompresi *file* PDF denganmenggunakan algoritma *Arithmetic Coding* sebesar 54%.

3.3 Hasil Pengujian

Perbedaan antara file PDF sebelum kompresi dan setelah kompresi dapat dilihat dari hasil pengujian di bawah ini :

Tabel 7. Hasil Pengujian Kompresi

No	Sebelum Dikompresi		Sesudah Dikompresi			
	Nama *File* PDF	Ukuran	Nama *File* PDFTerkompresi	Ukuran	Kinerja	
1	File_PDF_1.pdf	4,36 MB	File_PDF_1.ac	2,35 MB	RC CR Rd	1,667 60% 40%
2	File_PDF_2.pdf	5,05 MB	File_PDF_2.ac	3,69 MB	RC CR Rd	1,739 57,5% 42,5%
3	File_PDF_3.pdf	7,15 MB	File_PDF_2.ac	4,57 MB	RC CR Rd Rd	1,667 60% 40% 41,2%

Dari hasil pengujian di atas, dapat disimpulkan bahwa ukuran file PDF yang dikompresi lebih kecil dibandingkan dengan file PDF yang tidak dikompresi

4. KESIMPULAN

Berdasarkan hasil pengujian dapat disimpulkan bahwa ukuran file PDF yang dikompresi lebih kecil dibandingan dengan file PDF yang tidak dikompresi. Setelah mengikuti proses kompresi dengan algoritma Arithmetic Coding, file PDF yang berukuran cukup besar dapat dikompresi menjadi file PDF baru yang berukuran lebih kecil. Setelah menerapkan algoritma Arithemtic Coding untuk mengompresi file PDF, terbukti file PDF berhasil dikompres, dengan hasil Compression Ratio (RC) = 2,15 Compression Ratio (CR)) = 46% dan Redudancy (Rd) = 54 %.

REFERENCES

[1] R. Tari, S. D. Nasution, and T. Zebua, "Penerapan Algoritma ArithmeticCoding Untuk Mengkompresi Record Database Pada Aplikasi Ensiklopedia Flora Berbasis Android," vol. 2, no. 7, 2021.
[2] H. T. Sihotang, "PERANCANGAN DAN IMPLEMENTASI ALGORITMA ARITHMETIC CODING UNTUK APLIKASI KOMPRESI DATA VIDEO DAN AUDIO," vol. 2, no. 1, pp. 58–64, 2018.
[3] F. A. Sianturi, "Kompresi File Citra Digital Dengan Arithmetic Coding," vol. 03, pp. 45–51, 2018.
[4] R. A. Purba and L. Sitorus, "ANALISIS PERBANDINGAN ALGORITMA ARITHMETIC CODING DENGAN ALGORITMA LEMPEL ZIV WELCH (LZW) DALAM KOMPRESI TEKS," vol. 03, pp. 158–165, 2018.
[5] D. Salomon and G. Motta, "Handbook of Data Compression 5th," J. Chem. Inf. Model., vol. 53, no. 9, pp. 1689–1699, 2019.
[6] R. Ariska, "Penerapan Algoritma Arithmatic Coding Pada Aplikasi Kamus Teknologi Informasi Berbasis Android," vol. 2, no.

Jurnal Sains Dan Teknologi Informasi
Vol 3, No 2, Maret 2024, Hal 53-60
ISSN 2809-610X (Media Online)
DOI 10.47065/jussi.v3i2.4884
Website https://ejurnal.seminar-id.com/index.php/jussi

7, pp. 407–413, 2021.
[7] N. L. W. S. R. Ginantara et al., BASIS DATA : Teori dan Perancangan. 2020.
[8] E. Hariska et al., "Perancangan Aplikasi Kompresi File Gambar Menggunakan Algoritma Additive Code," vol. 5, hal. 193–202, 2021, doi: 10.30865/komik.v5i1.3671.
[9] D. Salomon dan G. Motta, Handbook of Data Compression 5th, vol. 53, no. 9. 2019.
[10] R. Wanti, "Rancang Bangun Sistem Informasi Akademik Pada SMK Citra Dharma Berbasis JAVA," J. Teknol. Inf., vol. 5, no. 2, hal. 86–92, 2019.
[11] A. N. A. Andrias, "Peranan Sistem Informasi Manajemen Pai Untuk Meningkatkan Kualitas Guru Di Smp Plus Al Munawar," Textura, vol. 1, no. 1, hal. 1–12, 2020.
[12] S. R. Saragih dan D. P. Utomo, "Penarapan Algoritma Prefix Code Dalam Kompresi Data Teks," KOMIK (Konferensi Nas. …, vol. 4, hal. 249–252, 2020, doi: 10.30865/komik.v4i1.2691.
[13] A. Apijuddin Kompresi dan A. S. Codes, "Perancangan Aplikasi Kompresi File Teks Menggunakan Algoritma Stout Codes," vol. 9, hal. 183–188, 2021.