



## Analisis Perbandingan Algoritma Run Length Encoding Dan Algoritma Five Modulus Dalam Kompresi File PDF

**Cindy Nadia Ambarita**

Universitas Budi Darma, Indonesia, email: cindyambarita84@gmail.com

### Info Artikel

**Diajukan:** 18-01-2024  
**Diterima:** 18-01-2024  
**Diterbitkan:** 21-01-2024

#### Kata Kunci:

Perbandingan; Run Length Encoding (RLE); Five Modulus; Kompresi; PDF; Analisis

#### Keywords:

Comparison; Run Length Encoding (RLE); Five Modulus; Compression; PDF; Analysis



Lisensi: cc-by-sa

Copyright © 2023 by Author. Published by Faatuatua Media Karya

### Abstrak

Perkembangan teknologi terkini mempengaruhi ukuran file, terutama file berformat Pdf. File Pdf cukup besar untuk menggunakan banyak ruang penyimpanan. File dokumen Portable Document Format (PDF) merupakan salah satu software yang digunakan untuk menyimpan data yang bersifat rahasia maupun biasa. Dalam sebuah sistem informasi masalah keamanan dan menjaga kerahasiaan data merupakan salah satu aspek yang penting. Untuk melakukan ini, maka diperlukan solusi seperti pembuatan media penyimpanan yang berkapasitas besar. Namun ke depannya, ukuran file Pdf akan bertambah dan kualitas Pdf akan meningkat, sehingga solusi ini menjadi kurang efektif. Oleh karena itu, diperlukan solusi tambahan untuk menghemat ruang media penyimpanan dan mengurangi ukuran data yang disimpan. Mengurangi penggunaan memori dan ukuran file tanpa mempengaruhi informasi dalam file dapat diatasi dengan kompresi. Pada perbandingan kedua algoritma Run Length Encoding dengan algoritma Five Modulus sebagai berikut, pada algoritma Run Length Encoding ini bekerja dengan baik pada kompresi file Pdf, dan dapat mencapai rasio kompresi yang sangat tinggi.

### Abstract

Recent technological developments affect file sizes, especially PDF format files. Pdf files are large enough to take up a lot of storage space. Portable Document Format (PDF) document files are a type of software used to store confidential and ordinary data. In an information system, security issues and maintaining data confidentiality are important aspects. To do this, a solution is needed, such as creating large capacity storage media. However, in the future, the size of PDF files will increase and the quality of PDFs will increase, so this solution will become less effective. Therefore, additional solutions are needed to save storage media space and reduce the size of stored data. Reducing memory usage and file size without affecting the information in the file can be achieved with compression. In comparing the two Run Length Encoding algorithms with the Five Modulus algorithm as follows, the Run Length Encoding algorithm works well in PDF file compression, and can achieve a very high compression ratio.

## 1. PENDAHULUAN

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, data atau file untuk menghasilkan repesntasi digital yang padat yang masih mengandung atau mewakili jumlah informasi yang terkandung dalam data [1]. 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.

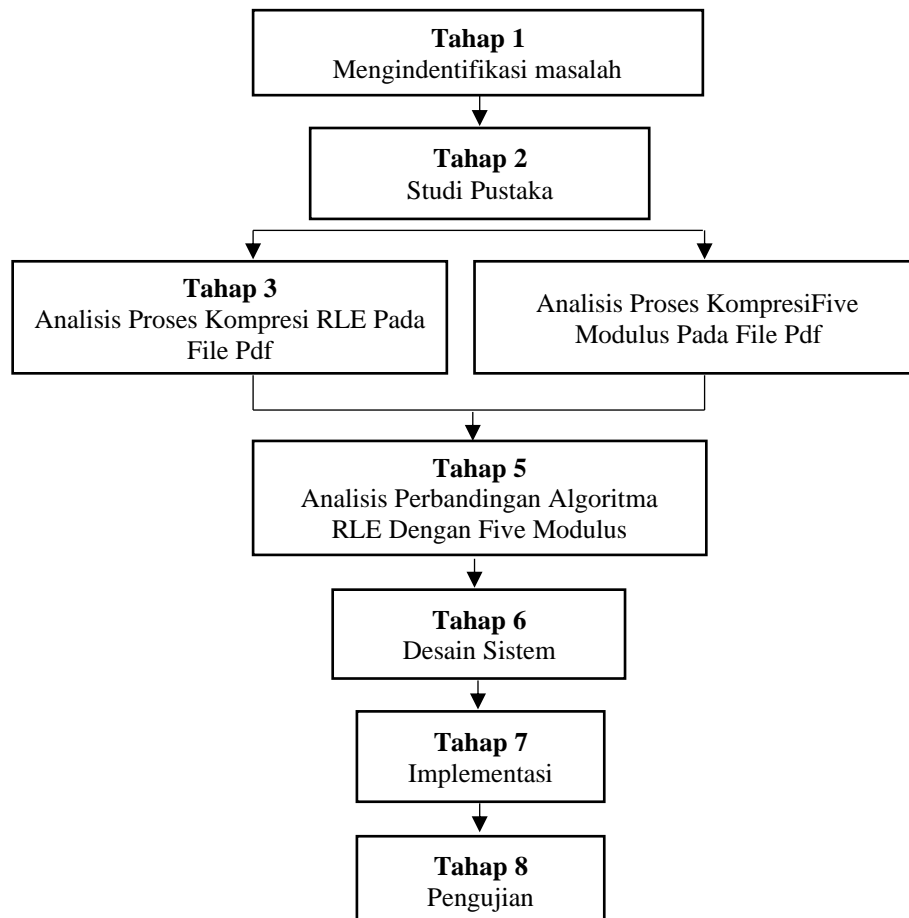
Jenis - jenis kompresi data ada dua yang pertama kompresi data lossless dan kompresi data lossy. Yang dimana kompresi data lossless yaitu kompresi data yang digunakan untuk mengompres file tanpa kehilangan kualitas dan informasi dari file aslinya. Sederhananya dalam kompresi data lossless, ukuran file berkurang, tetapi kualitas datanya tetap sama [2]. Algoritma – algoritma yang dibahas oleh peneliti ini ada dua yaitu algoritma Run Length Encoding dan Algoritma Five Modulus, yang pertama algoritma run length encoding ialah salah satu jenis lossless yang paling sederhana dari skema kompresi data dan didasarkan pada prinsip sederhana data encoding run length encoding sangat sesuai digunakan untuk mengompresi data yang berisi karakter-karakter berulang atau data berjalan (yaitu: urutan dimana nilai file akan sama terjadi pada banyak elemen data akan berurutan [3].

Pada penelitian terdahulu oleh Bernita siburian pada tahun 2020 dengan judul “Implementasi Five Modulus Untuk Kompresi File Audio”, digunakan untuk pemampatan data audio sehingga kompresi dapat dilakukan untuk mencapai hasil pemuatan metode tersebut [1]. Pada penelitian terdahulu oleh Umar Mansyuri pada tahun 2021 dengan judul “Kompresi Data Teks Dengan Metode Run Length Encoding”, dalam penelitian nya dapat menghitung jumlah kemunculan simbol lalu menuliskan simbol tersebut sebanyak satu kali diikuti dengan jumlah kemunculannya [2]. Pada penelitian terdahulu oleh Ulfa Lu’luilmaknum dan Nilza Humaira Salsabila pada tahun 2017 dengan judul “Penggunaan Metode Run Length Encoding Untuk Kompresi Data”, dalam penetician nya kompresi data dapat dikembangkan oleh suatu program run length encoding yang membantu proses kompresi data seperti file audio, MP3 ataupun data yang lainnya [3].

## 2. METODE PENELITIAN

### 2.1 Kerangka Penelitian

Kerangka penelitian adalah jenis atau bentuk kerangka kerja yang dapat digunakan sebagai pendekatan untuk memecahkan suatu masalah yang melibatkan langkah-langkah penelitian yang sistematis. Kerangka penelitian yang digunakan adalah sebagai berikut:



**Gambar 1.** Metode Penelitian

Dibawah ini beberapa metode dan teknik yang digunakan dalam pengumpulan dan pengolahan data untuk penelitian ini antara lain:

1. Identifikasi masalah  
Langkah ini adalah bagaimana penulis memprediksi, memperkirakan, dan mengkarakterisasi sumber masalah pada penyimpanan bervolume tinggi sehingga tidak dapat lagi disediakan di area penyimpanan yang ada.
2. Studi Pustaka  
Pada saat ini, penulis sedang mencari referensi dan literatur tentang teori-teori yang relevan dengan pertanyaan yang akan diajukan dalam penelitian ini. Referensi yang akan digunakan berupa buku, jurnal, artikel, tesis dan sumber lain yang berkaitan dengan masalah yang akan penulis bahas pada penelitian selanjutnya.
3. Tahap Analisis Proses Kompresi RLE Pada File Pdf  
Sampai disini kompresi file PDF sudah selesai yaitu kompres file PDF menggunakan algoritma Run Length Encryption dan mempelajari teknik pemrograman dasar menggunakan bahasa pemrograman Microsoft Visual Studio 2010. Tahap Analisis
4. Proses Kompresi Five Modulus Pada File Pdf  
Sampai disini kompresi PDF sudah selesai yaitu mengkompresi file PDF menggunakan algoritma Five Modulus dan mempelajari teknik pemrograman dasar menggunakan bahasa pemrograman Microsoft Visual Studio 2010.
5. Analisis Perbandingan Algoritma RLE Dengan Five Modulus  
Selama proses ini, dilakukan perbandingan antara algoritma enkripsi run-length dan algoritma lima modul, untuk menemukan algoritma terbaik menurut tiga parameter kompresi proporsional, kompresi proporsional, dan ekonomi ruang.
6. Desain Sistem  
Pada tahap ini diwujudkan ide dan desain sistem aplikasi yang dapat memodelkan data yang diperoleh.
7. Implementasi  
Pada tahap ini, implementasi desain aplikasi data pdf dilakukan dengan menggunakan bahasa pemrograman komputer.
8. Pengujian  
Langkah ini dilakukan untuk mengecek hasil kompresi file pdf. Proses ini dimaksudkan untuk menentukan hasil akhir dari penelitian yang dilakukan. diimplementasikan.

## 2.2 Algoritma RLE (*Run Length Encoding*)

Algoritma RLE (*Run Length Encoding*) Sebagai salah satu algoritma yang digunakan untuk mengkompresi data menjadi ukuran yang lebih kecil dari ukuran sebelumnya, algoritma ini berguna pada data dengan banyak nilai yang mirip secara berurutan seperti file gambar dan animasi, dan lain-lain selain algoritma Metode ini tidak cocok untuk data normal karena akan menghasilkan ukuran data yang lebih besar dari data aslinya.[6]

Algoritma RLE (*Run Length Encoding*) merupakan teknik kompresi sederhana yang mengurangi ukuran data digital dengan menghilangkan nilai yang berulang, teknik ini bekerja dengan mengidentifikasi urutan nilai data yang identik secara berurutan dan menggantinya dengan jumlah berapa kali nilai tersebut muncul.

Secara umum, algoritma RLE secara optimal digunakan pada file yang memiliki karakter yang cenderung homogen. Karena itu, jika algoritma digunakan secara universal maka perlu dilakukan pengelompokan atau transformasi karakter / simbol yang serupa. Langkah-langkah mengompresi data menggunakan *Run Length Encoding* adalah sebagai berikut:

1. Mengurangi ukuran data yang mengandung simbol berulang.
2. Simbol berulang ditandai sebagai run
3. 1 run biasanya dikodekan menjadi data 2-byte.
4. Byte kedua mewakili simbol berulang, yang disebut run value.
5. Byte pertama mewakili jumlah simbol, yang disebut run count.
6. RLE cocok untuk mengompresi data teks yang berisi banyak pengulangan simbol

## 2.3 Algoritma Five Modulation

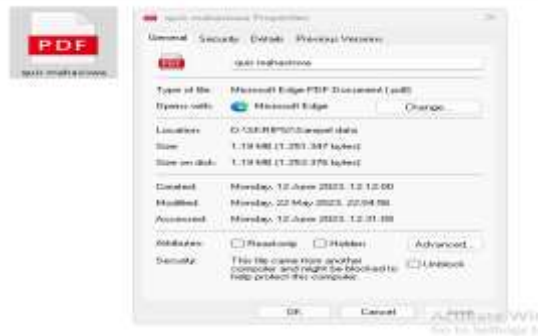
Algoritma Five Modulation adalah teknik kompresi dengan mengatur setiap nilai bit menjadi kelipatan 5 untuk setiap bit. Dalam proses kompresi five modulus data yang dikompresi adalah sampel Pdf dalam bentuk aliran bit yang terletak di bagian data Pdf. Five Modulus bekerja dengan mengurangi ukuran kata yang sering muncul dan meningkatkan bit yang jarang muncul.[1]

Algoritma five modulus adalah metode kompresi Pdf yang menyesuaikan setiap nilai bit dalam kelipatan 5 untuk setiap hexa. Kemudian nilai hexa dapat dibagi 5 untuk mendapatkan nilai baru untuk setiap hexa dan mengurangi di memori asli. Dan protokol baru untuk kompresi dari nilai pdf baru untuk kompresi yang mudah. Prosedur kompresi metode five modulus terdiri dari 3 langkah yaitu hexa(n) dimodulus 5, hexa(n) dibagi 5 dan hexa(n) dikurangi hexa terkecil. Untuk modulus proses 5,. dilakukan dengan algoritma berikut:

1. jika  $A(i, j) \text{ Mod } 5 = 4$
2.  $A(i, j) = A(i, j) + 1$
3. Berbeda jika  $A(i, j) \text{ Mod } 5 = 3$
4.  $A(i, j) = A(i, j) + 2$
5. Berbeda jika  $A(i, j) \text{ Mod } 5 = 2$
6.  $A(i, j) = A(i, j) - 2$
7. Berbeda jika  $A(i, j) \text{ Mod } 5 = 1$
8.  $A(i, j) = A(i, j) - 1$
9. Setelah menyelesaikan modul 5, hasil hex(n) dibagi 5..
10. Hasil pembagian hex(n) direduksi menjadi hex(n).
11. Kemudian hasil dari semua bit akan diubah menjadi biner dengan panjang 6 hex.
12. Setelah semua hasil dikonversi ke biner, biner digabungkan menjadi satu.
13. Setelah biner ditambahkan, itu akan dibagi 8 bit.
14. Kemudian, sebelum mendapatkan hasil keseluruhan dari kompresi akhir, ditambahkan string yaitu padding bit dan flag bit.

## 2.4 Sampel Data

Melaksanakan kompresi File Pdf, maka kita akan melakukan analisis pada File pdf. Berdasarkan batasan masalah diatas maka file Pdf yang akan di kompres adalah file yang berisi karakter string (teks), selain itu tidak bisa, Sebelum mengompres file Pdf pastikan terlebih dahulu file yang akan di kompres. Dalam hal ini, file Pdf akan dikompresi dengan nama sampel data Pdf seperti yang ditunjukkan di bawah ini:



Gambar 2. Sampel Data pdf



Gambar 3. Nilai Hexa Pdf

Tabel 1. Nilai hexadesimal sebelum dikompresi

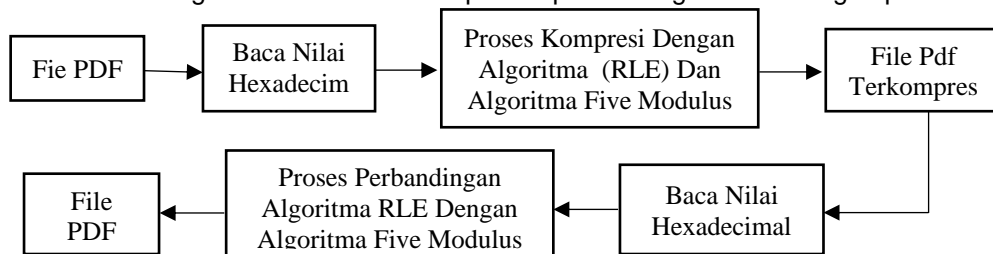
Hexa	Biner	Bit	Frekuensi	Bit * Frekuensi
B5	10110101	8	4	32
0D	00001101	8	2	16

25	00100101	8	2	16
0A	00001010	8	1	8
2D	00101101	8	1	8
2E	00101110	8	1	8
31	00110001	8	1	8
35	00110101	8	1	8
44	01000100	8	1	8
46	01000110	8	1	8
50	01010000	8	1	8
Bit Total				128 t

### 3. HASIL DAN ANALISIS

Analisis dari penelitian ini adalah menghitung dan merancang sebuah perangkat lunak untuk kompres file pdf, dimana penulis membandingkan dua algoritma yaitu algoritma Run Length Encoding dengan algoritma Five Modulus, untuk mengetahui kualitas dari masing-masing algoritma. Algoritma tipe pdf yang akan dikompresi menjadi MB Algoritma run-length encoding dan algoritma lima modul merupakan teknik kompresi lossless, yang dapat memperkecil ukuran data sesuai dengan karakter yang terdapat pada objek.gambar yang akan dikompresi. File megabyte (mb) adalah unit informasi digital yang dikirim atau disimpan, banyak digunakan di komputer dan teknologi informasi. Saat mem-parsing file MB, hal yang harus dilakukan adalah mendapatkan sampel file MB dengan membaca file MB. Pembacaan file MB dilakukan untuk mendapatkan nilai data dalam file MB sebagai bilangan heksadesimal melalui aplikasi Hxd.

Berikut adalah diagram sederhana untuk proses perbandingan saat mengompresi file PDF.



Gambar 5. Prosedur kompresi dan Perbandingan Pada file pdf

#### 3.1 Penerapan Algoritma Run Length Encoding

Sebelum File pdf dikompresi, maka terlebih dahulu mengambil sampel data pdf dan nilai hexa, setelah mendapatkan nilai hex dari file pdf, langkah selanjutnya menghitung data contoh diatas dalam langkah RLE yaitu dengan menyajikan data diatas menjadi tabel. Misalnya, 16 heksadesimal digunakan.

Tabel 2. Nilai hexadesimal sebelum dikompresi

Hexa	Biner	Bit	Frekuensi	Bit * Frekuensi
B5	10110101	8	4	32
0D	00001101	8	2	16
25	00100101	8	2	16
0A	00001010	8	1	8
2D	00101101	8	1	8
2E	00101110	8	1	8
31	00110001	8	1	8
35	00110101	8	1	8
44	01000100	8	1	8
46	01000110	8	1	8
50	01010000	8	1	8

Bit Total	128 Bit
-----------	---------

Nilai heksadesimal (Letter) memiliki nilai 8-bit dari bilangan biner. Untuk membuat 16 bilangan heksadesimal memiliki nilai biner 128 bit, untuk melakukan ini coba ubah satuannya menjadi byte hingga jumlah total bit dibagi 8. Hingga hasilnya  $128/8 = 16$  Setelah bilangan heksadesimal diurutkan berdasarkan kejadiannya frekuensi dan nilai biner diperoleh, kemudian menjalankan komputasi pengkodean panjang berikutnya mengurutkan bit dan menerima bit file yang dikompresi. Ada juga hasil kompresi file pdf yang dapat dilihat pada Tabel 4.4 di bawah ini.

**Tabel 3.** Nilai hexadesimal Sesudah dikompresi Dengan Run Length Encoding (RLE)

No	Nilai Hexadecimal	Nilai Biner	Bit	Frekuensi	Bit * Frekuensi
1	B5	10101	5	4	20
2	0D	1101	4	2	8
3	25	00101	5	2	10
4	0A	1010	4	1	4
5	2D	101101	6	1	6
6	2E	101110	6	1	6
7	31	110001	6	1	6
8	35	0110101	7	1	7
9	44	1000100	7	1	7
10	46	1000110	7	1	7
11	50	1010000	7	1	7
Total Bit					88 Bit

Berdasarkan tabel, maka ukuran akhir dari string "B5 0D 25 0A 2D 2E 31 35 44 46 50 " adalah 88 bit. Setelah perhitungan selesai, rasio kompresi akan diperoleh dan kita dapat melihat bagaimana rasio pengurangan ukuran diperoleh. Untuk mengetahui seberapa efektif algoritma kompresi, editor akan mengukur hasil kompresi PDF sesuai dengan parameter yang ditentukan. Semakin kecil file terkompresi yang dihasilkan, semakin tinggi kualitas kompresinya dan sebaliknya. Ukuran sebelum kompresi 128 bit, Ukuran setelah kompresi 88 bit

Parameter berikut digunakan untuk mengukur kinerja algoritma RLE ( Run Length Encoding).

1. Ratio of Compression (RC)

$$Rc = \frac{\text{ukuran data sebelum Dikompresi}}{\text{Ukuran data setelah dikompresi}}$$

$$Rc = \frac{128}{88} = 1,454$$

2. Cmpression Ratio (CR)

$$CR = \frac{\text{Ukuran data setelah dikompresi}}{\text{Ukuran data sebelum dikompresi}} \times 100$$

$$CR = \frac{88}{128} \times 100\% = 68,75\%$$

3. Space Saving (Ss)

$$Ss = 100\% - Cr$$

$$Ss = 100\% - 68,75\%$$

$$Ss = 31,25\%$$

Perhitungan di atas dapat disimpulkan bahwa rasio file Pdf yang dikompres dengan algoritma Run Length Encoding adalah 31.25%

Keterangan:

Setelah proses kompresi selesai, maka proses kompresi menggunakan algoritma run length encoding telah selesai maka hasil kompresinya bisa kita lihat dibawah ini:

Ukuran awal : 1,19 Megabyte

Compresi Ratio : 68,75%

Ukuran setelah dikompresi : 810 Kilobyte

### 3.2 Penerapan Algoritma Five Modulus

Berdasarkan contoh nilai hex file Pdf di atas, diambil 16 sampel nilai hex untukkompresifilepdf.

**Tabel 4.** Sampel nilai PDF dari bilangan heksadesimal

25	50	44	46	2D	31	2E	35	0D	0A
25	B5	B5	B5	B5	0D				

Saat menyelesaikan masalah berdasarkan ukuran kapasitas file pdf, algoritma lima modul diimplementasikan untuk mengompres file pdf sehingga dapat membaca bit.

**Tabel 5.** Pembacaan Nilai Hexa

Sampel	Frekuensi
25	2
50	1
44	1
46	1
2D	1
31	1
2E	1
35	1
0D	2
0A	1
B5	4

Mengurutkan karakter yang memiliki frekuensi kemunculan terbanyak hingga terkecil maka urutan hexadecimal dapat dilihat pada tabel dibawah ini.

**Tabel 6.** Pengurutan Frekuensi

No	Nilai Hexadecimal	Frekuensi
1	B5	4
2	0D	2
3	25	2
4	0A	1
5	2D	1
6	2E	1
7	31	1
8	35	1
9	44	1
10	46	1
11	50	1

Mendapat nilai file maka pencarian dilakukan dengan cara mengalihkan jumlah bit bilangan biner dengan frekuensi kemunculan dapat dilihat pada tabel dibawah ini.

**Tabel 7.** Nilai Bit Sampel PDF

No	Hexa	Ascii	Biner	Bit	Frekuensi	Bit * Frekuensi
1	B5	181	10110101	8	4	32
2	0D	13	00001101	8	2	16
3	25	21	00010101	8	2	16
4	0A	10	00001010	8	1	8
5	2D	45	00101101	8	1	8
6	2E	46	00101110	8	1	8
7	31	25	00011001	8	1	8
8	35	29	00011101	8	1	8

9	44	68	01000100	8	1	8
10	46	70	01000110	8	1	8
11	50	80	01010000	8	1	8
Total Bit						128

Mengukur dispersi nilai, maka dapat menghitung standar deviasi dalam tabel. Setelah algoritma FFM diterapkan dan mengubah nilai menjadi kelipatan 5 yang baru, dapat dilihat pada tabel dibawah ini.

**Tabel 8.** Transformasi Menggunakan Algoritma Five Modulus

No	Hexadecimal
1	181
2	13
3	21
4	10
5	45
6	46
7	25
8	29
9	68
10	70
11	80

Setelah ditransformasi menggunakan algoritma ffm, maka blok tersebut harus dibagi menjadi blok yang baru dengan 5 ( $a:b=c$ ) sehingga didapatkan hasil seperti di bawah ini.

**Tabel 9.** Tabel Hasil Bagi 5

No	Hexadecimal
1	36
2	3
3	4
4	2
5	9
6	9
7	5
8	6
9	14
10	14
11	16

Pada tabel 9 dapat kita lihat bahwa standar deviasi pada tabel lebih asli dari blok swap, yang berarti bahwa dispersi antar data menjadi lebih sedikit setelah transformasi menggunakan algoritma lima modul. Hasilnya adalah blok-blok baru tersebut sangat berkorelasi, yaitu datanya berkurang dibandingkan dengan data asli karena mereka berdekatan, yang akan lebih sedikit tersebar.

Selain itu, blok minimum baru yang ditunjukkan pada Tabel 7 dapat ditemukan dan dikurangkan dari semua blok. Berdasarkan tabel 7, minimalnya adalah 1 dan 4, jadi nilai terkecil yang kita gunakan adalah 1 untuk nilai yang sama dengan 1 dan 4 untuk nilai yang lebih besar dari 1, setelah dikurangkan satu blok kita mendapatkan blok baru.

**Tabel 10.** Hasil bagi 5

No	Hexadecimal
1	36-1
2	3-1
3	4-1
4	2-1
5	9-1
6	9-1
7	5-1
8	6-1
9	14-1
10	14-1
11	16-1

Sesudah menjumpai nilai yang akan dikurangi dengan nilai minimal, maka nilai maksimumnya dapat dilihat pada tabel dibawah ini.

**Tabel 11.** Tabel Setelah Blok Dikompresi

No	Hexadecimal
1	35
2	2
3	3
4	1
5	8
6	8
7	4
8	5
9	13
10	13
11	15

Setelah kita lihat pada tabel 11. bahwa angka teratas ialah 35 dan representasi dalam biner coding adalah (100011) yang panjangnya 6 bit.

**Tabel 12.** Bit Sesudah Dikompresi

No	Hexa	Bit	Frekuensi	Bit * Frekuensi
1	13	010011	2	12
2	8	001000	2	12
3	35	110101	1	6
4	2	000010	1	6
5	3	001100	1	6
6	1	000001	1	6
7	4	000100	1	6
8	5	001101	1	6
9	15	010101	1	6
Total Bit				66

Perhitungan di atas berhenti ketika diperoleh nilai yang lebih kecil dan diperoleh nilai biner, dan keenam bilangan biner tersebut digabungkan menjadi satu kesatuan sehingga menjadi:00001011 00001000 00110101 00000010 00000011 000000001 00000100 00000101 00010101

Berdasarkan perhitungan dengan algoritma five modulus di atas, terdapat 72 bit, kemudian bagi hasilnya menjadi 8 bit untuk masing-masing kelompok. Kemudian sebelum mendapatkan hasil keseluruhan dari kompresi akhir dilakukan penambahan urutan bit itu sendiri yaitu bit padding dan bit flag. Jika sisa panjang bit string 8 adalah 0, komplemennya adalah 00000001. Ekspresikan dengan bit terakhir. Sedangkan jika sisa panjang bitstring pada 8 adalah  $n(1,2,3,4,5,6,7)$  maka tambahkan 0 sampai  $7 - n + "1"$  pada akhir bit string. Nyatakan dalam L. Kemudian tambahkan bilangan biner dari 9 ke  $n$  dan nyatakan dengan bit terakhir. Karena nomor string 72 bit tidak habis dibagi 8 dan sisanya adalah 6 bit, nyatakan sisa pembagian dengan nilai  $n$ . Kemudian tambahkan  $7 - n + "1"$  0 ke akhir string bit. Bicaralah dengan L Kemudian jumlahkan bilangan biner dari 7 sampai  $n$ . Deklarasikan dengan bit terakhir.

$$7 - n + "1"$$

$$7 - 2 + "1" = 01$$

Gigitan terakhir  $9 - n$   
 Gigitan terakhir =  $9 - 2 = 7 = 00000111$

**Gambar 6.** Perhitungan Penambahan Bit

00001011 00001000 00110101 00000010 00000011 000000001 00000100 00000101 00010101  
**00000111**

Total panjang bit setelah penambahan bit adalah  $72+8=80$ . Kemudian dilanjutkan dengan memisahkan bit menjadi beberapa kelompok. Setiap grup terdiri dari 8 bit seperti yang ditunjukkan di bawah ini:

00001011 00001000 00110101 00000010 00000011 000000001 00000100 00000101 00010101  
 00000111

Berlandaskan pada pengelompokan nilai biner, maka yang didapat yaitu 10 kelompok biner terkompresi baru dengan nilai biner dari bit yang ditambahkan. Setelah pembagian selesai, biner yang telah dipisah akan dimodifikasi dengan kode ASCII untuk mengetahui nilai string bit terkompresi. Mengenai nilai kompresi, dapat kita lihat pada tabel di bawah ini:

**Tabel 13.** Nilai desimal Dan Biner Kompresi

Urutan kompresi	Biner	Desimal	Karakter
1	00001011	11	
2	00001000	8	
3	00110101	53	5
4	00000010	2	
5	00000011	3	
6	00000001	1	
7	00000100	4	
8	00000101	5	
9	00010101	21	!
10	00000111	7	.

Maka "File Pdf" direpresentasikan sebagai "51!.", dan jika watak tersebut diangkat menggunakan aplikasi notepad, Berdasarkan keputusan kompresi dengan five modulus file

Pdf=000010110000100000110101000000100000001100000000100000100 00000101  
 00010101 00000111 = 80. Dengan dihitungnya kinerja kompresi yaitu:

Berikut parameter untuk mengukur kinerja algoritma RLE (Run Length Encoding) file awal sebelum dikompres ialah 128 bit, sehingga rasio kompresi nya adalah :

1. Ratio of Compression (RC)

$$Rc = \frac{\text{ukuran data sebelum Dikompresi}}{\text{Ukuran data setelah dikompresi}}$$

$$Rc = \frac{128}{80} = 1,60$$

2. Compression Ratio (CR)

$$CR = \frac{\text{Ukuran data setelah dikompresi}}{\text{Ukuran data sebelum dikompresi}} \times 100$$

$$CR = \frac{80}{128} \times 100\% = 62,5\%$$

3. Space Saving (Ss)

$$Ss = 100\% - Cr$$

$$Ss = 100\% - 62,5\%$$

$$Ss = 37,5\%$$

Prediksi diatas, maka dapat menyimpulkan bahwa dengan menggunakan five modulus dapat dikompresi sebanyak 62,5%. Setelah proses kompresi selesai, maka proses kompresi menggunakan algoritma five modulus telah selesai maka hasil kompresinya bisa kita lihat dibawah ini:

Ukuran awal : 1,19 Megabyte  
 Compresi Ratio : 62,5%  
 Ukuran setelah dikompresi : 740 Kilobyte

### 3.3. Analisis Perbandingan Algoritma Run Length Encoding dan Five Modulus Dalam Kompresi File Pdf

Tindakan yang harus dilakukan dalam menperkirakan serta mengumpamakan kedua algoritma tersebut yakni sebagai berikut:

1. Memastikan kualifikasi dibuat menyamakan kedua prosedur, ada pula kualifikasi tersebut merupakan Rasio kompresi(RC), Compression Ratio (CR), dan Ruang Saving (SS).
2. Pemberian angka dari tiap kualifikasi yang sudah diresmikan, angka ini diambil bersumber dari hasil analisa prosedur Run Length Encoding Serta Algoritma Five Modulus pada kompresi file.
3. Memastikan hasil ataupun prioritas keputusan bersumber pada nilai tiap kualifikasi, Hasil keputusan bisa di lihat pada bagan dibawah ini:

**Tabel 14.** Analisis Hasil Perbandingan Prosedur Kompresi

Prosedur	Ukuran awal	Cr	Hasil kompresi
Algoritma Run Length Encoding	1,19 Mb	68,75%	840 Kb
Algoritma Five Modulus	1,19 Mb	62,5%	740 Kb

## 4. KESIMPULAN

Kesimpulan dari penelitian yang telah diteliti, Berdasarkan hasil dari penerapan program yang telah dilaksanakan pada bab terdahulu, maka bisa disimpulkan bahwa:

1. Pada kompresi data menggunakan Algoritma Run Length Encoding Dan Algoritma Five Modulus telah berhasil dilakukan pada file pdf dengan membaca setiap string file PDF satu persatu.
2. Pada proses implementasi program proses kompresi mendapatkan hasil berupa parameter Rasio Kompresi, Rasio Kompresi dan pengaturan hemat waktu dengan menghitung setiap nilai pengurangan ukuran file sesuai rumus parameter yang sudah ditetapkan pada aplikasi.
3. Saat merancang aplikasi ia berhasil menerapkan Algoritma Run Length Encoding dan Algoritma Five Modulus untuk mengompresi file pdf dengan mengambil ukuran hasil kompresi.

## REFERENSI

- [1] B. Siburian, "Implementasi Five Modulus Untuk Kompresi File Audio," Pelita Inform. Inf. dan, vol. 9, pp. 23–31, 2020, [Online]. Available: <http://ejurnal.stmik-budidarma.ac.id/index.php/pelita/article/view/2735>
- [2] U. Mansyuri, "Kompresi Data Teks Dengan Metode Run Length Encoding," J. Ilm. Sist. Inf., vol. 1, no. 2, pp. 102–109, 2021, [Online]. Available: <http://simasi.lppmbinabangsa.id/index.php/home>
- [3] U. Lu'luilmaknun and N. H. Salsabila, "Penggunaan Metode Run Length Encoding Untuk Kompresi Data," Semin. Mat. Dan Pendidik. Mat. Uny 2017, no. 1, pp. 273–280, 2017.
- [4] B. A. B. Ii and T. Pustaka, "Implementasi algoritma..., Darwin Chandra," sssssp. 6–23, 2016.
- [5] A. Mathematics, "濟無No Title No Title No Title," pp. 1–23, 2016.
- [6] M. R. Ramadhan, "Analisa Perbandingan Algoritma Run Length Encoding Dengan Burrows-Wheeler Transform Dalam Kompresi File Video," vol. 6, no. November, pp. 322–332, 2022, doi: 10.30865/komik.v6i1.5716.

- [7] D. Sebagai, S. Satu, U. Memperoleh, and G. Sarjana, "IMPLEMENTASI KOMBINASI ALGORITMA KOMPRESI RUN LENGTH ENCODING DAN SHANNON-FANO PADA CITRA GRAYSCALE Nanda Nugraha DEPARTEMEN ILMU KOMPUTER / INFORMATIKA," 2016.
- [8] D. Willfrid, M. Simamora, G. Ginting, and Y. Hasan, "Implementasi Algoritma Run Length Encoding Pada Kompresi File Mp3," JURIKOM (Jurnal Ris. Komputer), vol. 3, no. 4, pp. 5–9, 2016.
- [9] F. N. S. Damanik, A. A. Lubis, B. E. Ezer, and H. W. Siregar, "Perbandingan Kompresi Citra Metode Five-Modulus dan Kuantisasi dengan Perbaikan Citra Histogram-Equalization," J. SIFO Mikroskil, vol. 18, no. 1, pp. 57–70, 2017, doi: 10.55601/jsm.v18i1.435.
- [10] A. P. Gilang Rizki Akbar, Rosa Andrie Asmara, "Analisis perbandingan algoritma rle dan huffman pada kompresi citra," Semin. Inform. Apl. 2019, vol. 1, no. 1, pp. 323–329, 2019.
- [11] N. Ngichones, "IMPLEMENTASI ALGORITMA FIVE MODULUS DALAM MENYEMBUNYIKAN DATA RAHASIA KE DALAM," no. April, 2018.