

Perancangan Aplikasi Kumpulan Novel Dengan Menerapkan Algoritma Fibonacci Code

Intan Maharani*, Muhammad Syahrizal, Kurnia Ulfa

Fakultas Ilmu Komputer Dan Teknologi Informasi, Prodi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
*Email: intanrani499@gmail.com

Abstrak- Ukuran File teks terkadang relatif besar dimana semakin baik kualitas file teks yang dihasilkan, maka ukuran file yang dibutuhkan untuk menyimpan file teks tersebut semakin besar. Dengan ukuran file teks yang sangat besar, pada saat melakukan proses pemindahan bisa saja proses pemindahan gagal karena media ruang penyimpanan melebihi batasnya. Adapun Solusi dalam permasalahan ini adalah bagaimana file teks tersebut dapat dikompresi guna untuk mempercepat pemindahan dan penyimpanan file teks. Kompresi pada file teks dilakukan dengan memperkecil ukuran file teks dengan proses mengurangi bit pada file teks, akan tetapi tidak menghilangkan data informasi didalamnya. Dengan melakukan kompresi, data yang besar akan berkurang ukurannya sehingga dapat menghemat ruang penyimpanan. Dalam penelitian ini, algoritma yang digunakan adalah Fibonacci code, dengan menggunakan algoritma tersebut, hasil kompresi dari nilai mempunyai hasil yang berbeda-beda dari setiap nilai nya, dan hasil kompresi akan menguntungkan dalam melakukan pengiriman, dan pemindahan file teks akan semakin mudah.

Kata kunci: Kompresi File Teks; Algoritma Fibonacci Code

Abstract-Text file sizes are sometimes relatively large where the better the quality of the resulting text files, the larger the file size needed to store the text files. With a very large text file size, when carrying out the transfer process the transfer process may fail because the media storage space exceeds the limit. The solution to this problem is how the text file can be compressed in order to speed up the transfer and storage of text files. Compression of text files is done by reducing the size of the text file by reducing the bits in the text file, but not eliminating the information data in it. By compressing, large data will be reduced in size so as to save storage space. In this study, the algorithm used is the Fibonacci code, by using this algorithm, the compression results of the values have different results for each value, and the compression results will be beneficial in sending, and moving text files will be easier.

Keywords: Text File Compression; Fibonacci Code Algorithm

1. PENDAHULUAN

Perkembangan teknologi saat ini menunjukkan begitu mudahnya mengakses sebuah informasi yang dibutuhkan dengan cepat [1]. Dengan banyaknya informasi yang terkumpul dan disimpan maka dibutuhkan media penyimpanan yang cukup besar [2]-[3]. Semakin bertambahnya dokumen dalam media penyimpanan maka butuh sebuah media yang mampu melakukan proses pengkompresian terhadap file dokumen yang akan disimpan sehingga dapat mengurangi kapasitas media penyimpanan

Algoritma kode Fibonacci merupakan salah satu jenis metode kompresi berbasis pengkodean dengan sistem kode universal dimana nilai bilangan bulat positif dari suatu data membentuk kata kode biner dengan memanfaatkan bilangan Fibonacci [4]-[5]. Untuk jenis metode kompresi dengan penyandian kode universal biasanya lebih cepat daripada metode yang menggunakan pohon biner seperti Huffman. Penggunaan Algoritma Fibonacci Code sebagai algoritma kompresi dijelaskan dalam jurnal Benny dan Stephen [6].

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

1. Studi Pustaka
Pada tahap ini dilakukan review terhadap buku, artikel, jurnal, dan hasil penelitian sebelumnya sebagai acuan yang diperlukan dalam melakukan penelitian. Kemudian untuk mendapatkan informasi terkait dengan Algoritma Fibonacci Code.
2. Analisis dan Perancangan Sistem
Menganalisa implementasi pengkompresian algoritma Fibonacci code untuk mengetahui nilai kapasitas kompresi file novel.
3. Pengujian dan Implementasi Sistem
Melakukan uji coba serta pengembangan apabila masih terdapat kekurangan, maka akan diperbaiki dan akan diimplementasikan ke dalam bentuk bahasa pemrograman visual menggunakan Microsoft Visual Studio Net 2008.

2.2 Rancangan

Desain atau desain didefinisikan sebagai proses penerapan berbagai teknik dan prinsip untuk tujuan mendefinisikan perangkat, proses atau sistem secara cukup rinci untuk memungkinkan realisasi fisiknya [7]. Untuk mengontrol proses desain, A. Davis mengusulkan serangkaian prinsip dasar dalam desain atau desain sebagai berikut:

1. Desain tidak boleh mengalami tunnel vision.
2. Desain tidak boleh diulang.
3. Desain harus terstruktur untuk mengakomodasi perubahan.

4. Desain harus terstruktur untuk menurunkan dengan benar, bahkan ketika data dan peristiwa menyimpang atau menghadapi operasi.
5. Desain bukan coding dan coding bukan desain.
6. Desain harus dinilai kualitasnya pada saat desain dibuat, bahkan setelah selesai.
7. Desain harus ditinjau ulang untuk meminimalkan kesalahan konseptual (semantik) [8].

2.3 Kompresi data

Pemampatan data dalam konteks ilmu komputer adalah ilmu atau seni dalam merepresentasikan informasi yang terkandung dalam data ke dalam bentuk yang lebih kompak (kecil). Perkembangan komputer dan multimedia mengakibatkan kompresi data menjadi sangat penting dan berguna dalam teknologi saat ini. Solomon (2007) mengemukakan definisi kompresi data adalah suatu proses yang mengubah suatu input berupa aliran data (sumber atau data asli mentah) menjadi aliran data lain (output, aliran bit, atau aliran data) [8].

2.3.1 Manfaat Kompresi

Beberapa manfaat kompresi data yang dapat dijabarkan adalah sebagai berikut :

1. Waktu pengiriman data pada saluran komunikasi data lebih singkat. Contohnya pengiriman Aplikasi melalui handphone, dan download internet dan lain-lain.
2. Untuk membutuhkan ruang memori dalam storage yang lebih sedikit dibandingkan dengan file yang tidak dimampatkan.

2.4 Algoritma Kode Fibonacci

Kode Fibonacci adalah kode panjang variabel, di mana bilangan bulat yang lebih kecil mendapatkan kode pendek. Kode diakhiri dengan dua bit satu, dan nilai yang diperoleh adalah jumlah dari nilai Fibonacci yang sesuai untuk bit yang ditentukan (kecuali bit terakhir, yang merupakan akhir kode) [9]. Langkah-langkah untuk membentuk kode Fibonacci adalah sebagai berikut:

1. Temukan bilangan bulat positif n yang lebih besar dari atau sama dengan
2. Temukan bilangan Fibonacci terbesar yang lebih kecil atau sama dengan n , kurangi nilai n dengan f dan catat sisa pengurangan nilai n dengan f .
3. Jika bilangan yang dikurangkan adalah bilangan pada deret Fibonacci $F(i)$, tambahkan angka "1" pada $i-2$ pada kode Fibonacci yang akan dibentuk.
4. Ulangi langkah 2, ubah nilai n dengan sisa pengurangan nilai n dengan f sampai nilai sisa pengurangan nilai n dengan f adalah 0.
5. Tambahkan angka "1" pada posisi paling kanan dari kode Fibonacci yang akan dibentuk.

Untuk memecahkan kode kode Fibonacci, hapus "1" paling kanan, lalu ganti dan tambahkan kode Fibonacci yang tersisa menggunakan deret Fibonacci. Contoh proses encoding dan decoding suatu bilangan menjadi kode Fibonacci dapat dilihat sebagai berikut: Nilai $n = 65$. Barisan Fibonacci yang paling dekat dengan nilai n adalah:

12 3 5 8 13 21 34 55

Proses pengurangan nilai n dengan setiap bilangan Fibonacci

$$65 - 55 = 10$$

$$10 - 8 = 2$$

$$2 - 2 = 0$$

Tabel 1. Proses Pembentukan Kode Fibonacci

Bilangan fibonacci dalam deret Fibonacci	F(0)	F(1)	F(2)	F(3)	F(4)	F(5)	F(6)	F(7)	F(8)	F(9)	F(10)
Bilangan Fibonacci	0	1	1	2	3	5	8	13	21	34	55
Kode Fibonacci sementara	-	-	0	1	0	0	1	0	0	0	1

Proses terakhir pembentukan kode Fibonacci adalah dengan menambahkan angka "1" pada posisi paling kanan dari kode Fibonacci sementara sehingga kode Fibonacci untuk angka 65 adalah "0100100011". Sedangkan untuk proses decoding, hilangkan angka paling kanan "1" pada kode Fibonacci, kemudian ganti angka 1 dengan angka Fibonacci secara berurutan dari kiri ke kanan dan jumlahkan semua hasil substitusinya. Jadi proses decoding untuk kode Fibonacci "0100100011" adalah sebagai berikut:

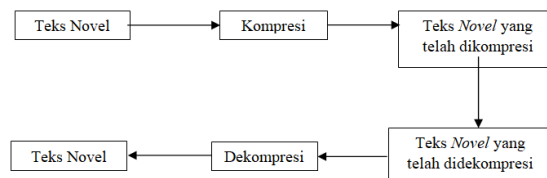
Dekode setelah "1" paling kanan dihilangkan: $0 \times F(2) + 1 \times F(3) + 0 \times F(4) + 0 \times F(5) + 1 \times F(6) + 0 \times F(7) + 0 \times F(8) + 0 \times F(9) + 1 \times F(10) = 0 + 2 + 0 + 0 + 8 + 0 + 0 + 0 + 55 = 65$. Terbukti bahwa bilangan 65 diubah menjadi kode Kode Fibonacci menjadi "0100100011", dan kode Fibonacci "0100100011" setelah decoding tetap menjadi angka 65.

Kode Fibonacci, seperti namanya, terkait erat dengan representasi bilangan bulat Fibonacci. Kode Fibonacci dari bilangan bulat positif n adalah representasi Fibonacci dari n dengan tambahan 1 di paling kanan. Jadi kode Fibonacci dari 5 adalah 0001|1 dan untuk 33 adalah 1010101|1. Sangat jelas bahwa kode diakhiri dengan sepasang "11" dalam kode.

Karena kode yang dihasilkan tidak memiliki dua angka 1 yang berurutan, kerapatan akan terlihat jelas dibandingkan dengan yang lain, jika beberapa nilai yang dicari adalah nilai kecil. Namun di sisi lain, hal ini juga membuat kode Fibonacci ini tahan terhadap kesalahan decoding [10].

3. HASIL DAN PEMBAHASAN

Pada analisa masalah ini, akan dibahas proses Sistem ini dirancang untuk melakukan proses kompresi pada novel. Proses kompresi menggunakan algoritma Fibonacci Codes dengan tujuan untuk mengetahui hasil performansi dari algoritma tersebut dalam mengompresi sebuah novel. Dan sistem ini dirancang menggunakan bahasa pemrograman Android.



Gambar 1. Proses Kompresi Dan Dekompresi Teks Novel

3.1 Penerapan Algoritma Fibonacci Codes

3.1.1 Kompresi Berdasarkan Algoritma Fibonacci Codes

Contoh kasus pada masalah ini adalah sebuah teks novel. Penulis akan melakukan percobaan kompresi pada teks novel tersebut. Teks novel yang akan dikompresi adalah “MENEMUKAN TILAMUTA DIMANAPUN”.

String : “menemukan tilamuta dimanapun”

Maka:

Bit Stringa : “menemukan tilamuta dimanapun”
 [string] : 28

Adapun langkah-langkah mengkompresi menggunakan algoritma fibonacci codes, yaitu:

- 1) Contoh proses kompresi teks novel dengan menggunakan algoritma fibonacci codes. Terdapat teks novel berisikan string “menemukan tilamuta dimanapun”.
- 2) Berdasarkan input string diatas, setiap karakter akan dipetakan kedalam tabel string yang belum dikompresi. Kemunculan karakter akan digambarkan didalam tabel 2:

Tabel 2. String Yang Belum Dikompresi

Char	ASCII Code	ASCII Code (Binary)	Bit	Frek	Bit*Frek
m	109	01101101	8	4	24
e	101	01100101	8	2	48
n	110	01101110	8	4	16
u	117	01110101	8	3	8
k	107	01101011	8	1	8
a	97	01100001	8	5	16
sp	32	00100000	8	2	16
t	116	01110100	8	2	8
i	105	01101001	8	2	8
l	108	01101100	8	1	8
d	100	01100100	8	1	8
p	112	01110000	8	1	8
Total				28	224

Berdasarkan kode ASCII, satu karakter bernilai delapan bit bilangan biner. Sehingga 28 karakter pada string mempunyai nilai biner sebanyak 160 bit. Sebelum melakukan proses kompresi, karakter tersebut diurutkan terlebih dahulu berdasarkan dari karakter yang memiliki frekuensi terkecil ke terbesar. Berikut merupakan contoh penyelesaian untuk mendapatkan Fibonacci Codes:

Penyelesaian :

Nilai $n = 1$, Deret fibonacci yang paling mendekati nilai n adalah :

0. 1. 1. 2. 3. 5. 8. 13. 21. 34. 55. atau Bilangan Fibonacci

Proses pengurangan nilai n dengan setiap bilangan fibonacci :

1. $n = 1$
 $fc = n - f$
 $= 1 - 1$
 $= 0$
 $Fc = 1$

| | | | | | | | | |
|---|---|---|---|---|----|----|----|----|
| 1 | 2 | 3 | 5 | 8 | 13 | 21 | 34 | 55 |
| 1 | - | - | - | - | - | - | - | - |

1 proses akhir pembentukan kode fibonacci adalah dengan menambahkan angka “1” pada posisi paling Kanan dari fibonacci menjadi “11”.

2. $n = 2$
 $fc = n - f$
 $= 2 - 2$
 $= 0$
 $Fc = 01$

| | | | | | | | | |
|---|---|---|---|---|----|----|----|----|
| 1 | 2 | 3 | 5 | 8 | 13 | 21 | 34 | 55 |
| 0 | 1 | - | - | - | - | - | - | - |

01 proses tamat pembentukan kode fibonacci adalah dengan menambahkan angka “1” pada posisi paling Kanan dari fibonacci menjadi “011”.

3. $n = 3$
 $fc = n - f$
 $= 3 - 3$
 $= 0$
 $Fc = 001$

| | | | | | | | | |
|---|---|---|---|---|----|----|----|----|
| 1 | 2 | 3 | 5 | 8 | 13 | 21 | 34 | 55 |
| 0 | 0 | 1 | - | - | - | - | - | - |

001 proses akhir pembentukan kode fibonacci adalah dengan menambahkan angka “1” pada posisi paling Kanan dari fibonacci menjadi “0011”.

4. $n = 4$
 $fc = n - f$
 $= 4 - 3$
 $= 1$
 $fc = 1 - 1$
 $= 0$
 $fc = 101$

| | | | | | | | | |
|---|---|---|---|---|----|----|----|----|
| 1 | 2 | 3 | 5 | 8 | 13 | 21 | 34 | 55 |
| 1 | 0 | 1 | - | - | - | - | - | - |

101 proses akhir pembentukan kode fibonacci adalah dengan menambahkan angka “1” pada posisi paling Kanan dari fibonacci menjadi “101”.

5. $n = 5$
 $fc = n - f$
 $= 5 - 5$
 $= 0$
 $Fc = 001$

| | | | | | | | | |
|---|---|---|---|---|----|----|----|----|
| 1 | 2 | 3 | 5 | 8 | 13 | 21 | 34 | 55 |
| 0 | 0 | 0 | 1 | - | - | - | - | - |

001 proses akhir pembentukan kode fibonacci adalah dengan menambahkan angka “1” pada posisi paling Kanan dari fibonacci menjadi “0011”.

6. $n = 6$
 $fc = n - f$
 $= 6 - 5$
 $= 1$
 $fc = 1 - 1$
 $= 0$
 $Fc = 1001$

| | | | | | | | | |
|---|---|---|---|---|----|----|----|----|
| 1 | 2 | 3 | 5 | 8 | 13 | 21 | 34 | 55 |
| 1 | 0 | 0 | 1 | - | - | - | - | - |

1001 proses akhir pembentukan kode fibonacci adalah dengan menambahkan angka “1” pada posisi paling Kanan dari fibonacci menjadi “10011”.

7. $n = 7$
 $fc = n - f$
 $= 7 - 5$
 $= 2$
 $fc = 2 - 2$
 $= 0$
 $Fc = 0101$

| | | | | | | | | |
|---|---|---|---|---|----|----|----|----|
| 1 | 2 | 3 | 5 | 8 | 13 | 21 | 34 | 55 |
| 1 | 0 | 1 | 0 | - | - | - | - | - |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | - | - | - | - | - |
|---|---|---|---|---|---|---|---|---|

0101 proses akhir pembentukan kode fibonacci adalah dengan menambahkan angka "1" pada posisi paling Kanan dari fibonacci menjadi "01011".

8. $n = 8$
 $fc = n - f$
 $= 8 - 8$
 $= 0$

$fc = 0001$

| | | | | | | | | |
|---|---|---|---|---|----|----|----|----|
| 1 | 2 | 3 | 5 | 8 | 13 | 21 | 34 | 55 |
| 0 | 0 | 0 | 0 | 1 | - | - | - | - |

0001 proses akhir pembentukan kode fibonacci adalah dengan menambahkan angka "1" pada posisi paling Kanan dari fibonacci menjadi "00011".

9. $n = 9$
 $fc = n - f$
 $= 9 - 8$
 $= 1$

$fc = 1 - 1$

$= 0$

$fc = 10001$

| | | | | | | | | |
|---|---|---|---|---|----|----|----|----|
| 1 | 2 | 3 | 5 | 8 | 13 | 21 | 34 | 55 |
| 1 | 0 | 0 | 0 | 1 | - | - | - | - |

10001 proses akhir pembentukan kode fibonacci adalah dengan menambahkan angka "1" pada posisi paling Kanan dari fibonacci menjadi "100011".

10. $n = 10$
 $fc = n - f$
 $= 10 - 8$
 $= 2$

$fc = 2 - 2$

$= 0$

$fc = 01001$

| | | | | | | | | |
|---|---|---|---|---|----|----|----|----|
| 1 | 2 | 3 | 5 | 8 | 13 | 21 | 34 | 55 |
| 0 | 1 | 0 | 0 | 1 | - | - | - | - |

010011 proses akhir pembentukan kode fibonacci adalah dengan menambahkan angka "1" pada posisi paling Kanan dari fibonacci menjadi "010011".

11. $n = 11$
 $fc = n - f$
 $= 11 - 8$
 $= 3$

$fc = 3 - 3$

$= 0$

$fc = 00101$

| | | | | | | | | |
|---|---|---|---|---|----|----|----|----|
| 1 | 2 | 3 | 5 | 8 | 13 | 21 | 34 | 55 |
| 0 | 0 | 1 | 0 | 1 | - | - | - | - |

00101 proses akhir pembentukan kode fibonacci adalah dengan menambahkan angka "1" pada posisi paling Kanan dari fibonacci menjadi "001011".

12. $n = 12$
 $fc = n - f$
 $= 12 - 8$
 $= 4$

$fc = 4 - 3$

$= 1$

$fc = 1 - 1$

$= 0$

$Fc = 10101$

| | | | | | | | | |
|---|---|---|---|---|----|----|----|----|
| 1 | 2 | 3 | 5 | 8 | 13 | 21 | 34 | 55 |
| 1 | 0 | 1 | 0 | 1 | - | - | - | - |

10101 proses akhir pembentukan kode fibonacci adalah dengan menambahkan angka "1" pada posisi paling Kanan dari fibonacci menjadi "101011".

Proses kompresi untuk Fibonacci Codes dapat dilihat pada Tabel 3.2

- 3) Kemudian lakukan pengurutan karakter secara descending kemudian bangkitkan Fibonacci Codes.
- 4)

Tabel 3. String Yang Sudah Dikompresi Dengan *Fibonacci Codes*

| N | Char | Fibonacci Codes | Bit | Frek | Bit*Frek |
|--------------|------|-----------------|-----|------|------------|
| 1 | a | 11 | 2 | 5 | 10 |
| 2 | m | 011 | 3 | 4 | 12 |
| 3 | n | 0011 | 4 | 4 | 16 |
| 4 | u | 1011 | 4 | 3 | 12 |
| 5 | sp | 00011 | 5 | 2 | 10 |
| 6 | e | 110011 | 5 | 2 | 10 |
| 7 | i | 01011 | 5 | 2 | 10 |
| 8 | t | 000011 | 6 | 2 | 12 |
| 9 | d | 100011 | 6 | 1 | 6 |
| 10 | k | 010011 | 6 | 1 | 6 |
| 11 | l | 001011 | 6 | 1 | 6 |
| 12 | p | 101011 | 6 | 1 | 6 |
| Total | | | | | 116 |

- 5) Setelah itu disesuaikan dengan karakter : m, e, n, e, m, u, k, a, n, sp, t, i, l, a, m, u, t, a, sp, d, i, m, a, n, a, p, u, n.
 "011, 10011, 0011, 10011, 011, 1011, 010011, 11, 0011, 00011, 000011, 01011, 001011, 11, 011, 1011, 000011, 11, 00011, 100011, 01011, 011, 11, 0011, 11, 101011, 1011, 0011".
- 6) jika sisa bagi panjang string bit terhadap 8 adalah 0, maka tambahkan "00000001", Nyatakan sebagai bit akhir.
- 7) jika terdapat sisa bagi n dari panjang string bit terhadap 8 (n= 1,2,3,4,5,6,7), maka tambahkan 0 sebanyak 7 - n + "1" nyatakan sebagai padding. Lalu tambahkan bilangan biner dari 9 - 4 = 5 00000101
 Padding = 7 - n + "1" = 7 - 4 + "1" = 0001
 Flagbits = biner 9 - n = 9 - 4 = 5 = 00000101
 Maka 116 + 12 = 128 bit adalah total bit setelah kompresi.

- 8) Selanjutnya melakukan perbandingan antara teks yang telah dikompres dan yang belum dikompres menggunakan rasio kompresi

a. Ratio of compression (Rc)

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

$$Rc = \frac{224 \text{ bit}}{128 \text{ bit}}$$

$$Rc = 1.75$$

b. Compression ration (Cr)

$$Cr = \frac{\text{Ukuran Data Setelah di Kompresi}}{\text{Ukuran Data Sebelum di Kompresi}} \times 100\%$$

$$Cr = \frac{128 \text{ bit}}{224 \text{ bit}} \times 100\%$$

$$Cr = 0.5714 \times 100\%$$

$$= 57.14\%$$

c. Rd

$$Rd = 100\% - \text{Compression ration (Cr)}$$

$$Rd = 100\% - CR$$

$$= 100\% - 57.14\%$$

$$Rd = 42.86\%$$

- 9) Selanjutnya biner tersebut dibagi menjadi 8 bit dan disesuaikan dengan nilai desimalnya:

| |
|---------------------------------|
| 01110011 = (115) = s |
| 00111001 = (57) = 9 |
| 10111011 = (187) = >> |
| 01001111 = (79) = O |
| 00110001 = (49) = 1 |
| 10000110 = (134) = + |
| 10110010 = (178) = ² |
| 11110111 = (247) = ÷ |
| 01100001 = (97) = a |
| 11100001 = (227) = ā |
| 10001101 = (141) = |
| 01101111 = (111) = I |
| 00111110 = (62) = > |
| 10111011 = (187) = >> |
| 00110001 = (50) = 2 |
| 00000101 = (5) = ENQ |

Berikut merupakan hasil kompresi dari teks novel : menemukan tilamuta dimanapun

s 9 >> O 1 + ² ÷ a ā I >>> 2 ENQ

3.1.2 Proses dekompresi berdasarkan algoritma Fibonacci code

Pada proses dekompresi, langkah awalnya sistem akan membaca string hasil kompresi, setelah mendapatkan string bit hasil kompresi, tabel karakter dan kodenya, kemudian sistem melakukan proses dekompresi dengan mengembalikan string bit tersebut kebentuk semula berdasarkan tabel karakter dan tabel kode yang telah didapat dari hasil kompresi.

- 1) Pada proses dekompresi hal yang dilakukan adalah menganalisa keseluruhan bit hasil dari kompresi sebelumnya. Adapun keseluruhan bit dan gabungan hasil kompresi yaitu:
 “01110011, 00111001, 10111011, 01001111, 00110001, 10000110, 10110010, 11110111, 01100001, 11100011, 10001101, 01101111, 00111110, 10111011, 00110001, 00000101”.
- 2) Proses awal dekompresi adalah dengan membaca input biner pada bit pertama yaitu “011” selanjutnya dilakukan pengecekan kedalam tabel fibonacci codes untuk mendapatkan karakter yang sesuai.

Tabel 4. Hasil Dekompresi Menggunakan *Fibonacci Codes*

| <i>Fibonacci Codes</i> | Karakter |
|------------------------|----------|
| 011 | m |
| 1011 | e |
| 0011 | n |
| 10011 | e |
| 011 | m |
| 1011 | u |
| 010011 | k |
| 11 | a |
| 0011 | n |
| 00011 | sp |
| 0000011 | t |
| 01011 | i |
| 001011 | l |
| 11 | a |
| 011 | m |
| 1011 | u |
| 000011 | t |
| 11 | a |
| 00011 | sp |
| 100011 | d |
| 01011 | i |
| 011 | m |
| 11 | a |
| 0011 | n |
| 11 | a |
| 101011 | p |
| 1011 | u |
| 0011 | n |

Berikut merupakan hasil dekompresi teks *novel* :

s 9 >> O 1 + ² ÷ a ā I >>> 2 ENQ

Menemukan tilamuta dimanapun

- 1) Baca nilai 8 bit akhir dari string bit dan ubah ke decimal maka 8 bit akhir dari string bit adalah 00000101 dengan nilai decimal 5, nyatakan dengan n.
- 2) Selanjutnya hapus bit pada akhir string bit sebanyak 7+n=7+5=12, maka 12 bit pada akhir string bit akan di hapus .
 “
- 3) “01110011, 00111001, 10111011, 01001111, 00110001, 10000110, 10110010, 11110111, 01100001, 11100011, 10001101, 01101111, 00111110, 10111011, 0011”.
- 4) Lakukan pembacaan bit pada awal string bit dan cek apakah terdapat pada tabel kompresi Fibonacci. Jika ada/terdapat maka ganti dengan karakter yang ada, jika tiadk ada maka pembacaan dilanjutkan dengan bit didepannya.

Tabel 5. pengecekan bit

| Indeks | Nilai | Keterangan |
|--------|-------|------------|
|--------|-------|------------|

| Indeks | Nilai | Keterangan |
|--------|--------|------------|
| 1 | 0 | kosong |
| 2 | 01 | kosong |
| 3 | 011 | Ada (m) |
| 4 | 1 | kosong |
| 5 | 10 | kosong |
| 6 | 100 | kosong |
| 7 | 1001 | kosong |
| 8 | 10011 | Ada (e) |
| 9 | 0 | Kosong |
| 10 | 00 | Kosong |
| 11 | 001 | Kosong |
| 12 | 0011 | Ada (n) |
| 13 | 1 | Kosong |
| 14 | 10 | kosong |
| 15 | 100 | kosong |
| 16 | 1001 | Tidak ada |
| 17 | 10011 | Ada (e) |
| 18 | 0 | kosong |
| 19 | 01 | kosong |
| 20 | 011 | Ada (m) |
| 21 | 1 | Kosong |
| 22 | 10 | kosong |
| 23 | 101 | kosong |
| 24 | 1011 | Ada (u) |
| 25 | 0 | kosong |
| 26 | 01 | kosong |
| 27 | 010 | kosong |
| 28 | 0100 | kosong |
| 29 | 01001 | kosong |
| 30 | 010011 | Ada (k) |
| 31 | 1 | kosong |
| 32 | 11 | Ada (a) |
| 33 | 0 | kosong |
| 34 | 00 | kosong |
| 35 | 001 | kosong |
| 36 | 0011 | Ada (n) |
| 37 | 0 | kosong |
| 38 | 00 | kosong |
| 39 | 000 | kosong |
| 40 | 0001 | kosong |
| 41 | 00011 | Ada (sp) |
| 42 | 0 | kosong |
| 43 | 00 | kosong |
| 44 | 000 | kosong |
| 45 | 0000 | kosong |
| 46 | 00001 | kosong |
| 47 | 000011 | Ada (t) |
| 48 | 0 | kosong |
| 49 | 01 | kosong |
| 50 | 010 | kosong |
| 51 | 0101 | kosong |
| 52 | 01011 | Ada (i) |
| 53 | 0 | kosong |
| 54 | 01 | kosong |
| 55 | 010 | kosong |
| 56 | 01011 | Ada (i) |
| 57 | 0 | kosong |
| 58 | 00 | kosong |
| 59 | 001 | kosong |
| 60 | 0010 | kosong |
| 61 | 00101 | kosong |

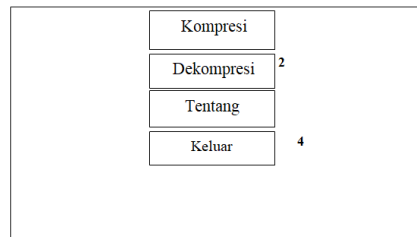
| Indeks | Nilai | Keterangan |
|--------|--------|------------|
| 62 | 001011 | Ada (l) |
| 63 | 1 | kosong |
| 64 | 11 | Ada (a) |
| 65 | 0 | kosong |
| 66 | 01 | kosong |
| 67 | 011 | Ada (m) |
| 68 | 1 | kosong |
| 69 | 10 | kosong |
| 70 | 101 | kosong |
| 71 | 1011 | Ada (u) |
| 72 | 0 | kosong |
| 73 | 00 | kosong |
| 74 | 000 | kosong |
| 75 | 0000 | kosong |
| 76 | 00001 | kosong |
| 77 | 000011 | Ada (t) |
| 78 | 1 | Tidak ada |
| 79 | 11 | Ada (a) |
| 80 | 0 | kosong |
| 81 | 00 | kosong |
| 82 | 000 | kosong |
| 83 | 0001 | kosong |
| 84 | 00011 | Ada (sp) |
| 85 | 1 | kosong |
| 86 | 10 | kosong |
| 87 | 100 | kosong |
| 88 | 1000 | kosong |
| 89 | 10001 | kosong |
| 90 | 100011 | Ada (d) |
| 91 | 0 | kosong |
| 92 | 01 | kosong |
| 93 | 010 | kosong |
| 94 | 0101 | kosong |
| 95 | 01011 | Ada (i) |
| 96 | 0 | kosong |
| 97 | 01 | kosong |
| 98 | 011 | Ada (m) |
| 99 | 1 | kosong |
| 100 | 11 | Ada (a) |
| 101 | 0 | kosong |
| 102 | 00 | kosong |
| 103 | 0001 | kosong |
| 104 | 00011 | Ada (n) |
| 105 | 1 | kosong |
| 106 | 11 | Ada (a) |
| 107 | 1 | kosong |
| 108 | 10 | kosong |
| 109 | 101 | kosong |
| 110 | 1010 | kosong |
| 111 | 10101 | kosong |
| 112 | 101011 | Ada (p) |
| 113 | 1 | kosong |
| 114 | 10 | kosong |
| 115 | 101 | kosong |
| 116 | 1011 | Ada (u) |
| 117 | 0 | kosong |
| 118 | 00 | Ada (d) |
| 119 | 001 | kosong |
| 120 | 0011 | Ada (n) |

3.2 Rancangan Antarmuka (Interface)

Perancangan interface adalah bagian yang penting dalam aplikasi, karena yang pertama kali dilihat ketika aplikasi dijalankan adalah tampilan antar muka (interface) aplikasi [11]. Perancangan antarmuka atau interface bertujuan untuk memudahkan interaksi antar manusia dengan computer sehingga manusia dapat menggunakan sistem yang telah dirancang dan dibangun dengan mudah [12].

1. Form Menu Utama

Pada form menu utama menampilkan button Menu Kompresi, Menu Dekompresi, Menu Bantuan, Menu Tentang dan Menu Keluar. Berikut ini pada gambar 3.6 merupakan rancangan menu utama.



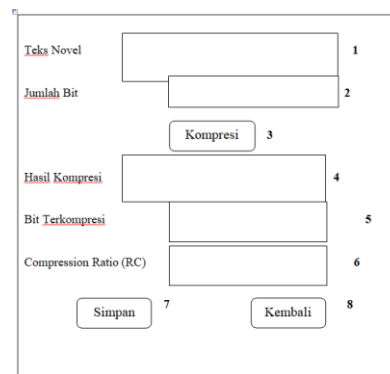
Gambar 3. Rancangan *Interface*

Keterangan:

1. Button Kompresi untuk menampilkan form proses kompresi teks novel.
2. Button Dekompresi untuk menampilkan form teks novel yang didekompresi.
3. Button Tentang untuk menampilkan keterangan tentang penulis, nama, npm, kelas, alamat, dan tanggal lahir.
4. Button Keluar untuk menampilkan keluar dari aplikasi kompresi, dekompresi, tentang pada Menu Utama.

2. Form Kompresi

Ketika user menekan tombol kompresi maka aplikasi ini akan menampilkan halaman untuk melakukan proses kompresi. Pada form inilah teks novel kemudian dikompresi untuk dikirim. Gambar rancangan dapat terdapat pada gambar dibawah ini.



Gambar 4. Rancangan *Form Kompresi*

Keterangan :

- 1) Edit Text Teks Novel untuk user menginputkan teks novel.
- 2) Edit Text Jumlah Bit untuk user menginputkan jumlah bit.
- 3) Button kompresi untuk mengkompresi teks novel.
- 4) Edit Text Hasil untuk menampilkan teks novel yang telah dikompresi.
- 5) Edit Text Bit Kompresi untuk menampilkan bit yang telah dikompresi.
- 6) Edit Text Compression Ratio (RC) untuk menampilkan rasio kompresi pada saat dikompresi
- 7) Button Simpan untuk menyimpan teks novel yang telah dikompresi.
- 8) Button Kembali untuk kembali ke menu utama.

3. Form Dekompresi

Button dekompresi menampilkan teks novel yang telah terkompresi, pada dasarnya form ini untuk dekompresi atau mengembalikan huruf yang telah terkompresi.

Gambar 5. Rancangan Form Dekompresi

Keterangan :

- 1) Edit Text Teks Novel Terkompresi untuk user menginputkan teks novel kompresi.
- 2) Edit Text Jumlah Bit Terkompresi untuk user menginputkan jumlah bit kompresi.
- 3) Button Dekompresi untuk mengembalikan teks novel seperti semula sebelum dikompresi.
- 4) Edit Text Hasil untuk menampilkan teks novel yang telah didekompresi.
- 5) Edit Text Bit Dekompresi untuk menampilkan bit yang telah didekompresi.
- 6) Button Simpan untuk menyimpan teks novel yang telah didekompresi.
- 7) Button Kembali untuk kembali ke menu utama.

4. Form Tentang

Untuk melihat tentang pembuat. Pada form ini, terdapat keterangan tentang penulis. Dapat dilihat pada gambar 6 berikut ini.

Gambar 6. Rancangan Form Tentang

4. KESIMPULAN

Kesimpulan yang dapat diambil setelah merancang aplikasi kompresi file baru adalah dengan menerapkan algoritma kode Fibonacci bahwa Prosedur kompresi file novel dimulai dengan memilih file novel yang akan dikompres, kemudian novel tersebut akan dikompres menggunakan algoritma sehingga dapat menghasilkan file kompresi terbaik dengan ukuran yang lebih kecil, Penerapan algoritma kode Fibonacci untuk kompresi file novel dilakukan dengan membaca nilai desimal hex dari file novel, kemudian mengubah nilai hex menjadi nilai bit baru yang berupa bilangan biner, kemudian menata ulang nilai biner tersebut menjadi karakter baru. Perancangan aplikasi kompresi file novel ini dilakukan dengan menggunakan Microsoft Visual Basic 2008 untuk mempermudah proses kompresi, hanya menginput file novel yang akan dikompres sehingga menghasilkan file kompresi beserta header-nya.

REFERENCES

- [1] R. D. Pratiwi, S. D. Nasution, and Fadlina, "Perancangan Aplikasi Kompresi File Teks Dengan Menerapkan Algoritma Fixed Length Binary Encoding (Flbe)," *Media Inform. Budidarma*, vol. 2, no. 1, pp. 10–14, 2018.
- [2] M. R. Irliansyah, S. D. Nasution, and K. Ulfa, "Penerapan Metode Deflate Dan Algoritma Goldbach Codes Dalam Kompresi File Teks," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 1, no. 1, pp. 186–189, 2017.
- [3] N. Dengen, "Perancangan Sistem Informasi Terpadu Pemerintah Daerah Kabupaten Paser," vol. 4, no. 1, pp. 47–54, 2009.
- [4] Marjito and G. Tesaria, "APLIKASI PENJUALAN ONLINE BERBASIS ANDROID (STUDI KASUS: DI TOKO HOAX MERCH)," *J. Comput. Bisnis*, vol. 10, no. 1, pp. 40–49, 2016.
- [5] D. Salomon and G. Motta, *Handbook of Data Compression*. Springer, 2010.
- [6] F. A. Hermawati, *Data Mining*. Yogyakarta: CV Andi Offset, 2018.
- [7] J. Enterprise, *Rahasia Manajemen File*. Jakarta: PT. Elex Media Komputindo, 2010.
- [8] A. Nugroho, *Rekayasa Perangkat Lunak Berorientasi Objek dengan Metode USDP*. Andi, 2010.
- [9] E. Prasetyo, *Data Mining Mengolah Data menjadi Informasi Menggunakan Matlab*. Yogyakarta: Andi, 2014.
- [10] R. A. S. M. Shalahuddin, *Rekayasa Perangkat Lunak*. Bandung, 2015.
- [11] A. Kadir, *REKAYASA PERANGKAT LUNA*. 2013.
- [12] M. Sadel, *7 Jam Belajar Interaktif Visual Basic.net 2008 Untuk Orang Awam*. Palembang: Maxikom, 2009.