

Penerapan Algoritma Fibonacci Code Pada File Transfer Berbasis Android

Dita Anggraini

Program Studi Teknik Informatika, Universitas Budi Dharma, Medan, Indonesia

Email: anggrainid046@gmail.com

Abstrak—Kompresi adalah mengecilkan sebuah ukuran data sehingga memori untuk mempresentasikan menjadi lebih sedikit daripada representasi data digital semula. Untuk Aplikasi File Transfer tidak membatasi jenis file. Ukuran data yang besar mengakibatkan pemborosan memori dan lambatnya proses pemindahan atau penduplikasian data sehingga perlu dilakukan cara untuk meminimalisir penyimpanan data. Ukuran file yang dikirim seperti gambar, audio, video dan dokumen. Jika file yang ditransfer ukurannya cukup besar maka perlu file tersebut dikompresi agar ukuran filenya mengecil dan rasionya 48%. Metode yang diterapkan untuk mengompresi aplikasi file transfer tersebut dengan menggunakan algoritma fibonacci code.

Kata Kunci: Kompresi; Fibonacci Code; File Transfer

Abstract—Compression is to reduce a data size so that there is less memory to present than the original digital data presentation. For File Transfer Applications do not limit file types. Large data size results in memory wastage and delays in the process of transferring or duplication of data so it needs to be done a way to minimize data storage. The size of files sent such as images, audio, video and documents. If the file transferred size is large enough then the file needs to be compressed so that the file size decreases and the ratio is 48%. The method applied to compress the file transfer application using the Fibonacci code algorithm.

Keywords: Compression; Fibonacci Code; File Transfer

1. PENDAHULUAN

Dunia teknologi berkembang dengan cepat dan semakin maju, tidak mudah dipungkiri kecanggihan teknologi banyak berperan dalam meringankan pekerjaan manusia. Salah satunya kebutuhan terhadap *transfer* atau pengiriman *file* yang semakin besar saat ini karena segala sesuatu sekarang bisa dilakukan hanya melalui *smartphone*. Pengguna *smartphone* sangat sering melakukan *transfer file*, baik berupa gambar, audio, video, dokumen, ataupun *file* sejenis lainnya. Saat ini banyak aplikasi *transfer file* yang dapat ditemukan di *smartphone* salah satunya *share it* dan aplikasi *file transfer* lainnya, yang sangat banyak ditemukan untuk melakukan mentransfer *file*.

Ukuran *file* yang ditransfer akan menjadi salah satu masalah jika ukuran suatu *file* tersebut sangat besar. Seiring data yang berkembang dan proses saat ini menjadi semakin besar dari waktu ke waktu mengakibatkan lamanya proses pengiriman karena ukuran *file* yang terlalu besar yang memungkinkan aplikasi tersebut lama mengirimkan *file*. Ukuran *file* yang besar juga menjadi masalah dalam penggunaan ruang penyimpanan pada sebuah *smartphone*.

2. METODOLOGI PENELITIAN

2.1 Transfer

Transfer menggunakan linguistik dan atau pengetahuan konseptual yang diperoleh sebelumnya untuk memudahkan sebuah tugas baru pembelajaran bahasa. *Transfer* adalah pengalihan performa atau pengetahuan sebelumnya ke dalam pembelajaran berikutnya. *Transfer* positif terjadi performa atau pengetahuan sebelumnya menunjang kegiatan pembelajaran selanjutnya. *Transfer* negatif terjadi ketika performa atau pengetahuan sebelumnya mengganggu performa pembelajaran sesudahnya [1].

2.2 Algoritma Fibonacci Code

Kode Fibonacci adalah variable length code, dimana bilangan bulat yang lebih kecil mendapatkan kode pendek. Kode berakhir dengan dua buah bit satu, dan nilai yang didapatkan adalah jumlah dari nilai-nilai Fibonacci yang sesuai untuk bit yang ditetapkan (kecuali bit terakhir, yang merupakan akhir dari kode). Langkah-langkah pembentukan sebuah kode Fibonacci adalah sebagai berikut:

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

8. Tambahkan angka “1” pada posisi paling kanan kode Fibonacci yang akan dibentuk.

Untuk melakukan decode sebuah kode Fibonacci, hilangkan angka “1” paling kanan, kemudian substitusi dan jumlahkan kode Fibonacci yang tersisa dengan menggunakan deret Fibonacci. Contoh proses encoding dan decoding sebuah bilangan menjadi kode Fibonacci dapat dilihat sebagai berikut:

Nilai $n = 65$. Deret Fibonacci yang paling mendekati nilai n adalah:

0 1 1 2 3 5 8 13 21 34 55 89

Proses pengurangan nilai n dengan setiap bilangan Fibonacci:

1. $65 - 55 = 10$
2. $10 - 8 = 2$
3. $2 - 2 = 0$

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 akhir pembentukan kode Fibonacci adalah 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 “1” paling kanan dalam kode Fibonacci, kemudian substitusi angka 1 dengan bilangan Fibonacci sesuai urutan dari kiri ke kanan dan jumlahkan seluruh hasil substitusi tersebut. Sehingga proses decode untuk kode Fibonacci “0100100011” adalah sebagai berikut:

Proses decode setelah angka “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 angka 65 diubah menjadi kode Fibonacci menjadi “0100100011”, dan kode Fibonacci “0100100011” setelah didecode tetap menjadi angka 65. Kode Fibonacci, sebagaimana namanya, sangat erat berhubungan dengan representasi bilangan bulat Fibonacci. Kode Fibonacci dari sebuah bilangan bulat positif n adalah representasi Fibonacci dari n dengan sebuah penambahan 1 diujung kanannya. Sehingga kode Fibonacci dari 5 adalah 00011 dan untuk 33 adalah 10101011. Sangat jelas bahwa kode-kode tersebut berakhir dengan sepasang angka “11” dalam kodenya. Karena kode yang dihasilkan tidak memiliki dua buah bit 1 yang berurutan, maka kepadatan akan terlihat jelas dibandingkan yang lain, apabila sebagian nilai yang dicari adalah nilai-nilai kecil. Namun disisi lain, ini juga membuat kode Fibonacci ini tahan terhadap kesalahan decode[2].

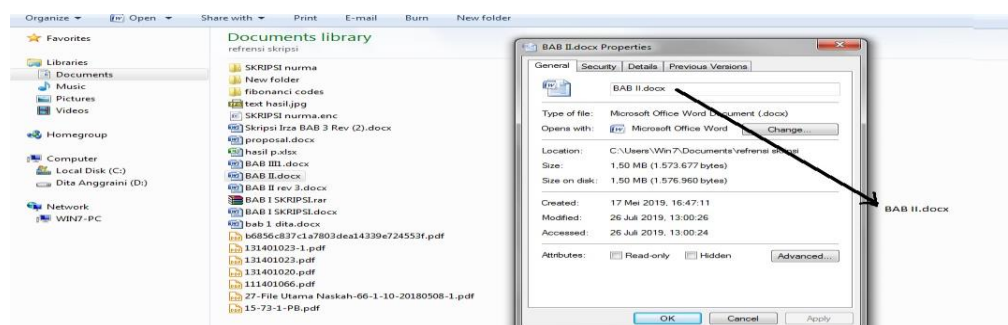
3. HASIL DAN PERANCANGAN

Berdasarkan penelitian ini, *file transfer* memiliki ukuran lebih besar dibanding ukuran *file* lainnya. Dengan melakukan kompresi data, data yang berukuran besar akan dikompresi menjadi ukuran yang kecil dan akan mengurangi alokasi penyimpanan. Dalam penelitian ini, akan dibahas 2 proses utama yaitu proses kompresi dan dekompresi, dan peneliti akan mengompresi sebuah *file transfer* dengan algoritma *fibonacci code*.

Contoh *file transfer* yang akan dikirim adalah *file docx* yang akan dilakukan sebuah proses percobaan kompresi pada *file docx*. Sebelum *file* dikompresi, terlebih dahulu dilakukan pembacaan pada *file docx* menggunakan aplikasi *Binary Viewer* untuk mencari biner pada aplikasi *file docx*. Berikut adalah contoh *file docx* yang akan dikompresi dan dekompresi.

Tabel 2. Contoh File Docx yang akan dikompresi

Nama File	BAB II.docx
Type	File Docx
Size	1.50 MB



Gambar 1. File Docx



3.2 Penerapan Algoritma *Fibonacci code*

Tabel 2. Nilai Simbol dan Kemunculan *frekuensi*

No	Simbol	Frekuensi
1	6E	2
2	74	2
3	6F	1
4	65	1
5	5F	1
6	54	1
7	79	1
8	70	1
Jumlah		10

Tabel 3. Nilai Hexsa (Karakter) Yang Belum Dikompresi

Simbol Hexa	Nilai Biner	Bit	Frek	Bit x Frek
6E	01101110	8	2	16
74	01110100	8	2	16
6F	01101111	8	1	8
65	01100101	8	1	8
5F	01011111	8	1	8
54	01010100	8	1	8
79	01111001	8	1	8
70	01110000	8	1	8
	Jumlah			80

JIMAT, Page 93

Tabel 4. Nilai Hexsa (Karakter) Dikompresi Dengan Nilai *Fibonacci Code*

N	Nilai Hexadesimal (Karakter)	Code word	Bit	Frek	Bit x Frek
1	6E	11	2	2	4
2	74	011	3	2	6
3	6F	0011	4	1	4
4	65	1011	4	1	4
5	5F	00011	5	1	5
6	54	10011	5	1	5
7	79	01011	5	1	5
8	70	000011	6	1	6
Total Bit					39

4. KESIMPULAN

Berdasarkan dari penelitian yang telah dilakukan, maka hasil akhir dari penelitian tersebut dapat diambil beberapa kesimpulan dari pembahasan sebelumnya. Adapun kesimpulan yang diperoleh prosedur pada aplikasi file transfer dalam mengirim sebuah file yaitu dengan memilih file yang akan dikompresi, setelah dipilih maka lakukan proses kompresi dan file hasil kompresi tersebut yang akan dikirim melalui jaringan, lalu proses dekompresi dilakukan oleh si penerima setelah file telah diterima. Penerapan algortima *Fibonacci Code* untuk mengkompresi *file* yang akan dikirim telah berhasil mmembuat ukuran ukuran file menjadi lebih kecil dengan rasio 48% sehingga dapat memberikan proses transmisi yang cepat serta membantu mengurangi pemakaian paket data internet. Aplikasi *file transfer* dirancang dan dibangun dengan menggunakan *tools eclipse juno* dan menerapkan algoritma *Fibonacci Code* sehingga dapat memperkecil ukuran file yang akan dikirim.

REFERENCES

- [1] J. I. Mulawarman *et al.*, "IMPLEMENTASI KRIPTOGRAFI PENGAMANAN DATA PADA PESAN TEKS , ISI FILE DOKUMEN , DAN FILE DOKUMEN MENGGUNAKAN ALGORITMA ADVANCED ENCRYPTION," vol. 10, no. 1, 2015.
- [2] Doroty L, Satoto K I, Nurhayati O D, PERANCANGAN IMPLEMENTASI SISTEM INFORMASI PERPUSTAKAAN DIPROGRAM STUDI TEKNIK LINGKUNGAN FAKULTAS TEKNIK UNDIP Vol.2, No.4, Oktober 2014 (e-ISSN:2338-0403)
- [3] M. K. Emy Setyaningsih, S.Si., *Kriptografi & Implementasinya menggunakan MATLAB*. Yogyakarta: ANDI, 2015.
- [4] Rifki Sadikin, *Kriptografi untuk keamanan jaringan dan implementasi dalam Bahasa Java*. Yogyakarta: ANDI, 2012