Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 3, Mei 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i3.2153

Implementasi Algoritma J-Bit Encoding Pada Kompresi File PDF

Frida Effelyanti Naibaho

Program Studi Teknik Informatika, STMIK Budi Darma, Medan, Indonesia
Email: effeliyantifrida@gmail.com

Abstrak–Kompresi merupakan pengurangan ukuran data menjadi ukuran yang lebih kecil dari aslinya. Adapun teknik dari kompresi ini adalah dengan mengganti karakter yang berulang-ulang tersebut dengan suatu pola tertentu sehingga data tersebut dapat meminimalisasi ukurannya. Oleh karena itu data-data yang akan disimpan perlu dikompres terlebih dahulu supaya ukurannya menjadi lebih kecil. Keunggulan dari file pdf adalah bahwa format file pdf tidak hanya dapat menyimpan data dalam bentuk teks saja namun juga bisa menyimpan sebuah gambar (image) atau pun foto. Dalam melakukan penyimpanan di dalam file pdf, data yang akan tersimpan tidak mudah terkena virus. Salah satu teknik kompresi file pdf yang dapat memampatkan citra dengan baik adalah kompresi file pdf menggunakan metode J-Bit Encoding. J-Bit Encoding (JBE) merupakan sebuah algorima yang mengoptimalkan masukan untuk algoritma kompresi lainnya, dengan demikian algoritma kompresi akan lebih baik jika dikombinasikan dengan metode J-Bit Encoding.

Kata Kunci: Kompresi, PDF, J-Bit Encoding.

Abstract–Compression is a reduction in the size of the data to a smaller size than the original. The technique of compression is to replace the repetitive character with a certain pattern so that the data can minimize its size. Therefore the data that will be stored needs to be compressed first so that the size becomes smaller. The advantage of pdf files is that the pdf file format can not only store data in text form but also can save an image (image) or photo. In storing in pdf files, the data that will be stored is not susceptible to viruses. One pdf file compression technique that can compress images well is compressing pdf files using the J-Bit Encoding method. J-Bit Encoding (JBE) is an algorithm that optimizes input for other compression algorithms, thus the compression algorithm will be better if it is combined with the J-Bit Encoding method.

Keywords: Compression, PDF, J-Bit Encoding.

1. PENDAHULUAN

Semakin berkembangnya teknologi pada komputer maka semakin banyak pula data-data atau file-file yang ingin kita simpan, tapi kadangkala kapasitas memory yang kita miliki tidak sebanding dengan data yang akan kita simpan. Oleh karena itu data-data yang akan disimpan perlu dikompres terlebih dahulu supaya ukurannya menjadi lebih kecil. Keunggulan dari file pdf adalah bahwa format file pdf tidak hanya dapat menyimpan data dalam bentuk teks saja namun juga bisa menyimpan sebuah gambar (image) atau pun foto. Dalam melakukan penyimpanan di dalam file pdf, data yang akan tersimpan tidak mudah terkena virus.

Masalah pada penyimpanan file pdf sering kali kita mengirimnya dengan kapasitas yang berukuran besar sehingga dalam mengirim file pdf dapat pemborosan pada pengiriman file pdf . Dengan demikian maka kapasitas ruang penyimpanan yang diperlukan akan lebih besar serta waktu dan ruang penyimpanan menjadi lebih besar pula. Solusi dalam mengatasi masalah tersebut harus di kompresi agar penyimpanan pada file pdf tersebut semakin menghemat dalam pengiriman file pdf. Dalam metode pengkompresian memiliki kelebihan dan kekurangan, kemampuan dari metode pengkompresian data yang diukur dengan ukuran atau kapasitas dan kecepatan kompresi. Dan file pdf yang telah dikompresi dapat dikirim dengan mudah dan lebih cepat karena kapasitas datanya lebih kecil.

Terdapat beberapa penelitian terkait dengan algoritma J-Bit Encoding, seperti yang dilakukan I Made Agus Dwi Suarjaya dengan judul penelitian A New Algorithm for Data Compression Optimization, hasil penelitian ini mengambil bentuk penelitian bagaimana algoritma J-Bit Encoding dikombinasikan dan dibandingkan dengan algoritma lain, dan dari hasil pengujian yang dilakukan algoritma J-Bit Encoding bisa dikombinasikan dengan baik untuk beberapa algoritma kompresi berbasis teks seperti LZW, LZ77, Run Length Encoding dan BurrowsWheeler[1].

Agar proses yang dilakukan lebih mudah, ada pun solusi yang ditawarkan dalam mengkompresi file pdf yaitu dengan menggunakan algoritma J-Bit Encoding. J-Bit Encoding merupakan algoritma kompresi data lossless yang memanipulasi setiap bit data dalam file untuk meminimalkan ukuran dengan cara membagi data menjadi dua keluaran, kemudian dikombinasikan kembali menjadi satu keluaran[2]. Kelebihan dari algoritma J-Bit Encoding adalah dapat menghasilkan rasio kompresi yang tinggi apabila data masukan memuat persentase byte nol yang tinggi. Sebaliknya, apabila persentase byte nol dari data masukan rendah, maka algoritme J-Bit Encoding akan menghasilkan file pdf kompresi yang dapat mendekati nilai satu, bahkan bisa melebihi satu (data terekspansi).

2. METODE PENELITIAN

2.1 Kompresi

Kompresi merupakan pengurangan ukuran data menjadi ukuran yang lebih kecil dari aslinya. Pengkompresian data ini sangat menguntungkan manakala terdapat suatu data yang berukuran besar dan data di dalamnya

Submitted: **21/04/2020**; Accepted: **16/05/2020**; Published: **20/05/2020**

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 3, Mei 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i3.2153

mengandung banyak pengulangan karakter. Adapun teknik dari kompresi ini adalah dengan mengganti karakter yang berulangulang tersebut dengan suatu pola tertentu sehingga data tersebut dapat meminimalisasi ukurannya. Masalah kompresi melibatkan proses bagaimana mengoptimalkan algoritma untuk menghapus berbagai redudansi dari suatu objek data. Terdapat beberapa permasalahan umum seperti jumlah bit total yang dikompresi, bagaimana mengembalikan ke bentuk string atau bit asli dan berapa banyak redundasi di ekstraksi dari data asli, menggunakan algoritma J-Bit Encoding (JBE) untuk melakukan kompresi terhadap file teks[1].

2.2 Algoritma J-BIT ENCODING

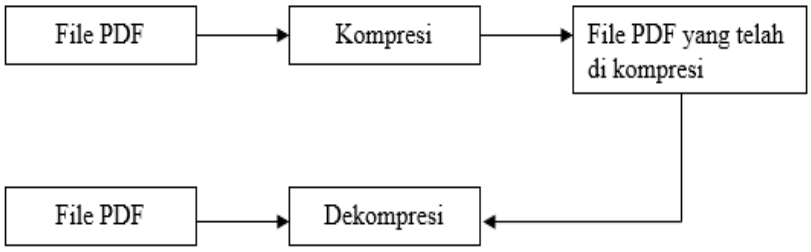
JBE merupakan algoritma kompresi data yang memanipulasi setiap bit data dalam file untuk meminimalkan ukuran tanpa kehilangan data apapun setelah proses decoding sehingga algoritma ini diklasifikasikan sebagai algoritma kompresi lossless Langkah-langkah proses kompresi dan dekompresi dari JBE dapat dilihat pada pseudocode berikut:

1. Proses Kompresi
   a. Baca masukan per byte
   b. Jika byte yang dibaca adalah nol maka tulis bit ‘0’ ke dalam temporary byte data. Jika byte yang dibaca bukan nol maka tulis byte tersebut pada data I dan tulis bit ‘1’ ke dalam temporary byte data.
   c. Ulangi langkah a dan b hingga temporary byte data terisi dengan 8 bit.
   d. Jika temporary byte data telah terisi dengan 8 bit maka tulis nilai byte dari temporary byte data tersebut ke dalam data II.
   e. Hapus temporary byte data.
   f. Ulangi langkah a sampai e hingga akhir file dicapai.
   g. Tulis kombinasi data I dan II
      1) Tulis panjang input asli.
      2) Tulis data I.
      3) Tulis data II.
   h. Jika diikuti oleh algoritma kompresi lain, data I dan data II dapat dikompresi secara terpisah sebelum digabungkan (opsional).
2. Proses Dekompresi:
   a. Baca panjang input asli.
   b. Jika dikompres secara terpisah, dekompresi data I dan data II (opsional).
   c. Baca Data II per bit.
   d. Jika bit yang dibaca adalah '1' maka baca data I dan tulis ke output, sebaliknya jika bit yang dibaca adalah '0' maka tulis byte nol ke output.
   e. Ulangi langkah c dan d hingga panjang input data asli dicapai.

3. HASIL DAN PEMBAHASAN

Analisa adalah tahap aktifitas kreatif dimana analis berusaha memahami permasalahan secara mendalam terhadap metode yang diterapkan ini adalah proses *interative* yang terus berjalan hingga permasalahan dapat dipahami dengan benar. Analisa bertujuan untuk mendapatkan pemahaman metode yang diangkat secara keseluruhan tentang sistem yang akan dibuat berdasarkan masukan dari pihak-pihak dan juga pengalaman analis yang berkepentingan dengan sistem tersebut.

Sistem ini dirancang untuk melakukan proses kompresi pada *file* PDF. Proses kompresi menggunakan algoritma *J-Bit Encoding* dengan tujuan untuk mengetahui hasil kinerja dari algoritma tersebut dalam mengkompresi sebuah *file* PDF. Adapun proses cara kerja kompresi penelitian ini adalah:



Gambar 1. Cara Kerja Proses Kompresi

Contoh kasus dari masalah ini adalah sebuah file yang berformat PDF akan dilakukan sebuah proses percobaan kompresi pada file PDF tersebut.

Sebelum file dikompresi terlebih dahulu dilakukan pembacaan pada file PDF untuk mendapatkan data berupa biner. Membaca biner yang terdapat pada file PDF menggunakan aplikasi Binary Viewer untuk mencari nilai biner pada aplikasi file PDF.

Submitted: **21/04/2020**; Accepted: **16/05/2020**; Published: **20/05/2020**

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 3, Mei 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i3.2153

Tabel 1. Sampel File PDF yang akan dikompresi

Nama File	FRIDA EFFELYANTI
Type	File PDF

Tabel 2. Karakter Ke Biner

No	Karakter	ASCII	Biner
1	F	70	01000110
2	R	82	01010010
3	I	73	01001001
4	D	68	01000100
5	A	65	01000001
6	SPACE	32	00100000
7	E	69	01000101
8	F	70	01000110
9	F	70	01000110
10	E	69	01000101
11	L	76	01001100
12	Y	89	01011001
13	A	65	01000001
14	N	78	01001110
15	T	84	01010100
16	I	73	01001001

Proses berikutnya adalah menentukan zero byte dan non zero byte dari setiap karakter dari kalimat nama file, berikut adalah proses kompresi untuk nama file.

Tabel 3. Proses Byte I

Original	Data I
01000110	0000000000000000
01010010	1111101111111111
01001001	0000010000000000
01000100	0100000000010010
01000001	0010000000110101
00100000	1001001111100110
01000101	1100000110000100
01000110	0010101001011001
01000110	
01000101	
01001100	
01011001	
01000001	
01001110	
01010100	
01001001	
128 bit	

Langkah berikutnya adalah menentukan nilai Bit 0 atau 1 yang dibaca untuk dimasukkan kedalam *temporary byte data*, dan pada kasus ini penulis memilih byte 1, maka proses berikutnya adalah menulis semua bit 1 kedalam *non zero byte* dan hasilnya sebagai berikut dengan:
Data I :
0000000000000000 1111101111111111 0000010000000000 0100000000010010 0010000000110101 1001001111100110 1100000110000100 0010101001011001 = 128 bit

Temporary Data Byte :
1111101111111111 00000100 0100000000010010 0010000000110101 1001001111100110 1100000110000100 0010101001011001 = **104 Bit**

Temporary Data Byte:
1111101111111111000001000100000000010010001000000011010110010011111001101100000110000100001010010011001

Kemudian *Temporary Byte Data* dibagi kedalam kelompok 8 bit untuk dimasukkan dan diproses kedalam Data II, dan hasilnya sebagai berikut:

Submitted: **21/04/2020**; Accepted: **16/05/2020**; Published: **20/05/2020**

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 3, Mei 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i3.2153

Tabel 4. Bit Data II

Temporary Byte Data	Data II
11111011 11111111	11111011
00000100 01000000	11111111
00010010 00100000	00000100
00110101 10010011	01000000
11100110 11000001	00010010
10000100 00101010	00100000
01011001	00110101
	10010011
	11100110
	11000001
	10000100
	00101010
	01011001

Data II merupakan hasil akhir dari proses kompresi dari kalimat ***FRIDA EFFELYANTI***, hasil biner tersebut dirubah kedalam bentuk *ASCII* akan menghasilkan *encoding* teks sebagai berikut : **û?@5IB*Y**, hasil kompresi menjadi 13 *byte* dan hasil kompresi lebih kecil dari *file original* 16 *byte*, proses *decoding* adalah mengembalikan proses kedalam bentuk asli, langkah awal adalah merubah û?@5IB*Y, kedalam bentuk biner dan hasilnya sebagai berikut:
11111011111111110000010001000000000100100010000000110101100100111110011011000001100001000010101001011001 .

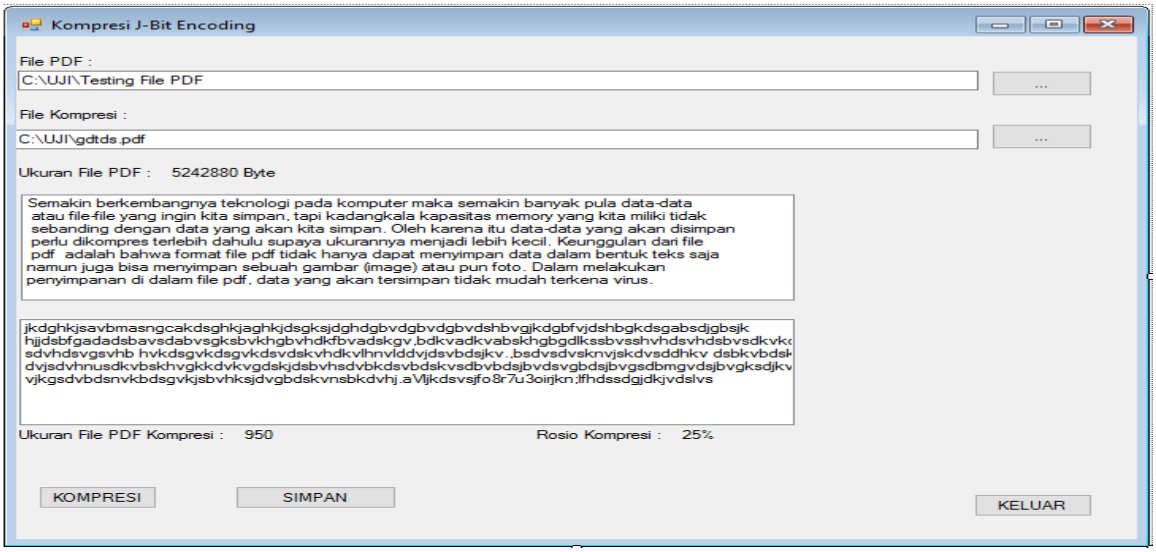
Kemudian berikutnya adalah merubah biner yang ada kedalam bentuk 16 bit dimulai dari sebelah kiri dan hasilnya sebagai berikut:
1111101111111111 0000010001000000 0001001000100000 0011010110010011 1110011011000001 1000010000101010 01011001.

Setelah di konversi dengan melakukan proses data *byte* I dan data *byte*II seperti pada proses kompresi, maka hasil biner ditambahkan *zero byte* dan hasilnya sebagai berikut:
0000000000000000 1111101111111111 0000010000000000 0100000000010010 0010000000110101 1001001111 100110 1100000110000100 0010101001011001.
Ketika proses konversi ke *ASCII* dilakukan akan mendapatkan hasil sebagai berikut: **FRIDA EFFELYANTI.**

3.1 Pengujian

Implementasi sistem program ini mencakup spesifikasi kebutuhan perangkat keras (*hardware*) dan spesifikasi perangkat lunak (*software*). Berikut ini perincian dari tampilan *output* dari perangkat lunak, Pengujian program digunakan untuk melihat hasil yang sudah dibahas, berikut adalah beberapa tampilan aplikasinya.

1. Pengujian Kompresi



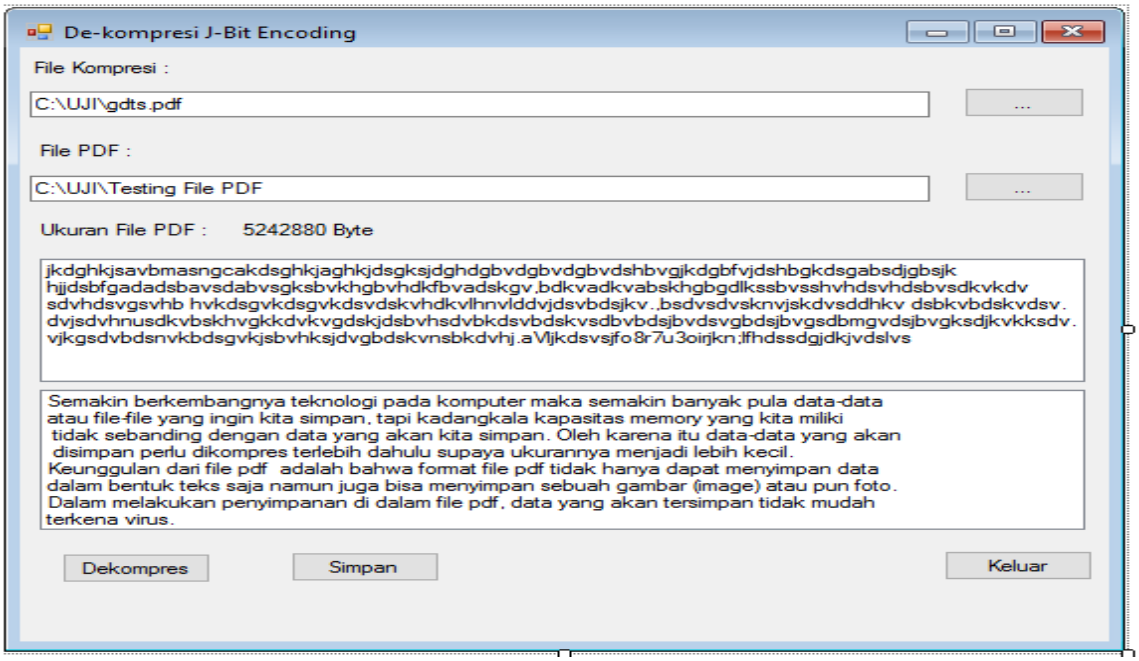
Gambar 2. Hasil Proses Pengujian Kompresi

Hasil kompresi diatas menampilkan proses kompresi dari ukuran asli 5242880 Byte menjadi 950 byte dengan ratio kompresi 25 %, ratio kompresi merupakan persentase kompresi yang berhasil dilakukan.

Submitted: **21/04/2020**; Accepted: **16/05/2020**; Published: **20/05/2020**

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 3, Mei 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i3.2153

2. Pengujian Dekompresi



Gambar 3. Hasil Pengujian Dekompresi

Pengujian kompresi dilakukan terhadap beberapa macam *file pdf* dengan konten yang berbeda, berikut adalah hasil pengujiannya.

Tabel 4. Hasil Pengujian

No	Jumlah Karakter	Ukuran Asli (Byte)	Hasil Kompresi (Byte)	Ratio Kompresi	Text
1	1231	1231	940	24%	Semakin berkembangnya teknologi pada komputer maka semakin banyak pula data-data atau file-file yang ingin kita simpan, tapi kadangkala kapasitas memory yang kita miliki tidak sebanding dengan data yang akan kita simpan. Oleh karena itu data-data yang akan disimpan perlu dikompres terlebih dahulu supaya ukurannya menjadi lebih kecil.
2	1094	1094	880	20%	The main objective of this study is to determine sound absorption characteristic of the acoustic material of fiber rod Roystonea regia by using polyurethane and gypsum as matrix. The variable in this study is the changes of the material fiber rod Roystonea regia size and composition in the matrix of polyurethane and gypsum. Preparation of sample is started by hewing, cleaving, drying, taking fiber, chopping, blending, sifting, mixing with polyurethane and gypsum, shaping the specimen and testing the Roystonea regia rod. The specimen then formed in cylindrical shape with 4 inchi diameter and 2 cm thick in order to be tested its density, porosity, sound absorption characteristic and the microstructure. The result of this study can be concluded that the highest absorption coefficient at a frequency 2000 Hz is 0.7966 is obtained from the fiber sized 32 mesh, with the composition 60%, the density is 0,4335 gr/cm3 and the porosity is 57,8791%. Whereas, the highest absorption coefficient at the fiber composition 50% is 0,6016, the density is 0,4511 gr/cm3 and the porosity is 55,6001%.

Submitted: **21/04/2020**; Accepted: **16/05/2020**; Published: **20/05/2020**

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 3, Mei 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i3.2153

4. KESIMPULAN

Dengan diselesaikannya penelitian ini yang mencakup penggunaan algoritma J-Bit Encoding dalam melakukan kompresi dapat diperoleh beberapa kesimpulan : Cara kerja algoritma J-Bit Encoding untuk mengkompresi file pdf yaitu membagi data masukan menjadi dua keluaran, kemudian di implementasi kembali menjadi satu keluaran.

REFERENCES

[1] T. Mizwar, G. L. Ginting, Mesran, A. Fau, S. Aripin, and D. Siregar, ”Implementasi Algoritma J-Bit Encoding Pada Kompresi File Teks,” KOMIK (Konferensi Nas .Teknol. Inf. dan Kompter), vol. 1, no. 1,pp.232-236,2017.
[2] J. K. M. Lobang, “Modifikasi Algoritma J-Bit Encoding untuk Meningkatkan Rasio Kompresi,”vol. 6, no. 1, 2017.
[3] V. Cipher, “Impelemtasi Vernam Cipher dan Steganografi End Of File (EOF) Untuk Enkripsi Pesan PDF,”vol. 15, no.1, pp 66-71,2016.
[4] Kartika Firdausy Balza Achmad, Pengolahan Citra Digital menggunakan Delphi. Yogyakarta: Penerbit Andi, 2012.
[5] Usman Ahmad, Pengolahan Citra Digital dan Teknik Pemrogramannya. Bogor: Penerbit Graha Ilmu, 2005.
[6] Edi Prihantoro, Elfitrin Syahrul Lulu C. Munggaran, "Penerapan Teknik Kompresi Burrows-Wheeler Pada Dokumen Berbahasa Indonesia" KNSI, p. 860, 2014.
[7] Giovanni Motta David Salomon, Handbook Data Compresion. New York: British Library Cataloguing, 2010.
[8] Marimin, Teknik Dan Aplikasi Pengambilan Keputusan Kriteria Majemuk. Jakarta, Indonesia: Grasindo, 2005.
[9] Adi Nugroho, Rekayasa Perangkat Lunak menggunakan UML dan JAVA. Yogyakarta: Penerbit Andi, 2009.
[10] S.Si., M.Kom. Emy Setyaningsih, Kriptografi dan Implementasinya menggunakan Matlab. Yogyakarta: Penerbit Andi, 2015
[11] Edi Prihantoro dan Elfitrin Syahrul Lulu C. Munggaran, "Penerapan Teknik Kompresi Burrows Wheller Pada Dokumen Berbahasa Indonesia," KNSI, vol. 175, no. KNSI2014-175, p. 922, Pebruari 2014..

Submitted: **21/04/2020**; Accepted: **16/05/2020**; Published: **20/05/2020**