



# ANALISIS KOMPRESI DATA TEKNIK *LOSSLESS* *COMPRESSION* MENGGUNAKAN DATA CALGARY CORPUS

Irwan Munandar  
Balai Pendidikan dan Pelatihan  
Tambang Bawah Tanah

## 1. Latar Belakang

Kompresi data merupakan suatu upaya untuk mengurangi jumlah bit yang digunakan untuk menyimpan atau mentransmisikan data[3]. Kompresi data meliputi berbagai teknik perangkat lunak (*software*) maupun perangkat keras (*Hardware*). Bila ditinjau dari sisi penggunaannya, kompresi data bisa bersifat umum untuk segala keperluan tertentu. Kompresi data bersifat khusus misalnya untuk data teks, data grafik, data suara, kode sumber (*Source program*) bahasa pemrograman tertentu, atau *database* suatu organisasi.

Secara umum kompresi data terdiri dari pengambilan simbol-simbol masukan dan mengubahnya menjadi kode-kode. Jika kompresi tersebut efektif maka kode-kode yang dihasilkan akan berukuran lebih kecil dari pada simbol-simbol asalnya. Keputusan mengubah simbol atau kumpulan simbol tertentu menjadi kode tertentu sehingga elemen-elemen dasar dari kompresi data adalah permodelan dan pengkodean[1].

Kompresi data dibedakan menjadi dua macam, yaitu : kompresi *Lossy* dan *lossless*[3]. Kompresi *lossy* memberi toleransi terhadap adanya loss pada data untuk menghasilkan rasio kompresi tinggi. Tingkat kualitas hasil kompresinya biasanya disesuaikan dengan batas kemampuan penilaian pengguna. Kompresi *lossy* biasanya diterapkan pada citra dan suara, sedangkan kompresi *lossless* menjamin *stream* data dari proses dekompresi akan tepat sama *stream* data aslinya. Kompresi bersifat *lossless* karena kehilangan 1 bit saja akan mengakibatkan data menjadi tidak berguna.

Tujuan dari penulisan artikel ini adalah untuk melakukan analisis kompresi data menggunakan algoritma BWT (*Burrows wheeler Transform*) dengan algoritma aritmatika. Pertanyaan penulis dalam artikel ini adalah “Algoritma kompresi data manakah dengan teknik *lossless compression* yang mampu menghasilkan rasio kompresi yang lebih baik?”. Teknik-teknik kompresi dan dekompresi data yang akan dibahas bersifat *Lossless compression* sehingga dapat diterapkan pada sembarang jenis file data.

## 2. Metode dan Landasan Teori

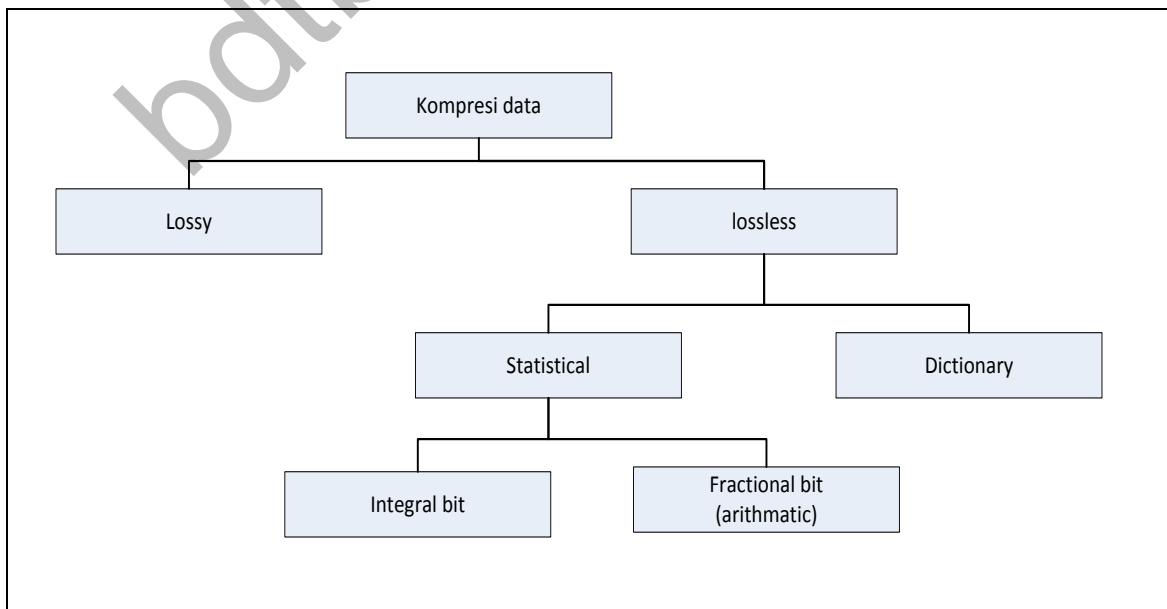
### 2.1 metode

Pada artikel ini penulis menggunakan alat bantu yaitu membangun aplikasi kompresi menggunakan bahasa pemrograman Delphi. sumber data untuk melakukan pengujian menggunakan beberapa *sample* file standar Calgary corpus yang memiliki karakteristik yang berbeda-beda, dilihat berapa ukuran awalnya untuk kemudian dilakukan proses kompresi dengan algoritma BWT dan algoritma aritmatika untuk dilihat rasio dan kecepatan prosesnya. Algoritma aritmatika yang digunakan pada evaluasi ini menggunakan aplikasi yang sama dengan algoritma BWT.

Proses pengambilan data dilakukan dengan beberapa metode yaitu studi kepustakaan yaitu pencarian materi pembahasan yang menyangkut artikel ini dengan mencari referensi dari beberapa buku dan membuat perancangan dengan menyusun konsep, membangun algoritma, dan menentukan struktur data yang akan dipakai.

### 2.2 landasan teori

kompresi data mempresentasikan data ke suatu bentuk kode lain yang lebih efisien atau berukuran lebih kecil dari ukuran aslinya tanpa menghilangkan makna penting dari data aslinya[3]. Dalam perkembangannya pembagian teknik kompresi data dapat digambarkan pada gambar 2.1.



Gambar 2.1 Pembagian teknik kompresi data[3]



Lossy compression merupakan kebalikan dari lossless compression dimana hasil kompresi jika di dekompresi tidak akan sama dengan data sebelum di kompresi. Lossy compression merupakan kompresi yang mengakibatkan hilangnya data-data tertentu atau keakuratan data untuk mencapai rasio yang lebih baik. Teknik ini biasanya diterapkan pada data suara atau citra digital. Dalam hal ini, alasan keakuratan data masih dapat diterima karena pada awalnya sendiri, data suara atau citra didapatkan dari hasil sampling dan kuantitas yang keduanya sendiri mengakibatkan data input dan output sudah tidak sama. Biasanya, pada lossy compression memiliki level atau tingkat kompresi yang dapat ditentukan yang tentunya juga berakibat pada seberapa akurat data hasil kompresi nantinya.

Lossless compression merupakan teknik kompresi yang menjamin data input dan output (hasil kompresi) adalah sama dari segi keakuratan, sehingga tidak boleh terjadi adanya kehilangan 1 bit saja. Teknik ini biasanya diterapkan pada teks, dokumen dan basis data. Lossless compression di implementasikan berdasarkan salah satu model berikut ini :

- a. model berbasis statistik, menggunakan dan membentuk kode simbol berdasarkan probabilitas kemunculan simbol tersebut, misalnya pada metode huffman.
- b. Model berorientasi karakter, menggunakan karakter khusus sebagai indikator kompresi, misalnya run length.
- c. Model berbasis kamus, mengkodekan simbol berdasarkan kode yang terdapat pada kamus, misalnya metode LZSS

### 2.2.1 Burrows wheeler Transform (BWT)

Transformasi Burrow wheeler (BWT) merupakan teknik pengurutan blok data yang dikembangkan oleh michael Burrows dan david Wheeler. Transformasi ini memproses sebuah blok data sebagai unit tunggal. Ide dasar dari BWT adalah dengan menerapkan sebuah transformasi, yang bersifat reversible, pada sebuah blok data untuk membentuk sebuah blok baru yang memiliki isi karakter yang sama, namun lebih mudah untuk di kompresi menggunakan dasar statistik[1].

Algoritma BWT merupakan suatu algoritma yang mengambil suatu blok data dan menyusunnya ulang menggunakan algoritma pengurutan. Blok hasil pengurutan memiliki elemen data yang sama dengan blok data awal, hanya saja urutannya yang berbeda[1]. Transformasi ini bersifat reversible yang artinya urutan asli dari elemen-elemen data dapat dikembalikan tanpa ada kehilangan (loss) pada keasliannya. Algoritma pertama merupakan algoritma transformasi kompresi, yang dilakukan pada blok teks sebelum proses kompresi, dan algoritma kedua menerangkan operasi inversnya[2].



### 2.2.2 Arithmetic Coding

Pengkodean aritmatik tidak menggunakan suatu kode spesifik untuk menggantikan suatu simbol tertentu. Arithmetic Coding menggunakan seluruh alirn simbol yang masuk dengan sebuah bilangan pecahan. Bilangan pecahan tersebut membutuhkan jumlah bit yang lebih banyak jika pesan yang dikodekan lebih panjang atau lebih kompleks. Hasil dari proses pengkodean aritmatik berupa sebuah bilangan pecahan yang lebih besar atau sama dengan nol dan lebih kecil dari satu. Bilangan ini dapat dikodekan kembali untuk menghasilkan simbol-simbol asalnya[3].

Penghitungan probabilitas pengkodean aritmatik yang diperoleh dengan bilangan pecahan dengan memunculkan setiap simbol yang telah di kodekan. Setelah itu ditetapkan suatu pembagian wilayah atau range untuk setiap simbol dari keseluruhan range antara nol dan satu sesuai dengan probabilitasnya. Dalam hal ini tidak ada ketentuan ada ketentuan bagian range mana yang harus diberikan simbol tertentu, asalkan program pengkodean dan pendekodean bagian yang sama untuk simbol tersebut[3].

Dalam proses kompresi data, simbol-simbol harus dikodekan sesuai dengan jumlah bit informasi yang dikandungnya. Karakter-karakter atau simbol-simbol tidak dapat dikedoken dengan menggunakan sistem ASCII atau EBCDIC karena sistem tersebut menggunakan jumlah bit yang sama untuk setiap karakter. Proses kompresi pada arithmetic coding yaitu dengan rumus berikut ini :

$$\text{Low} = \text{low} + \text{range} * \text{low\_range}(\text{character})$$

$$\text{High} = \text{low} + \text{range} * \text{high\_range}(\text{character})$$

$$\text{Range} = \text{high} - \text{low}$$

Dimana  $\text{low\_range}$  dan  $\text{high\_range}$  adalah batas bawah dan batas atas dari range setiap karakter yang diambil. Proses dekompresi pada arithmetic coding yaitu dengan rumus berikut ini :

$$\text{Range} = \text{high} - \text{low}$$

$$\text{Number} = \text{number} - \text{low\_range}(\text{character})$$

$$\text{Number} = \text{number} / \text{range}$$

### 3. Hasil dan Pembahasan

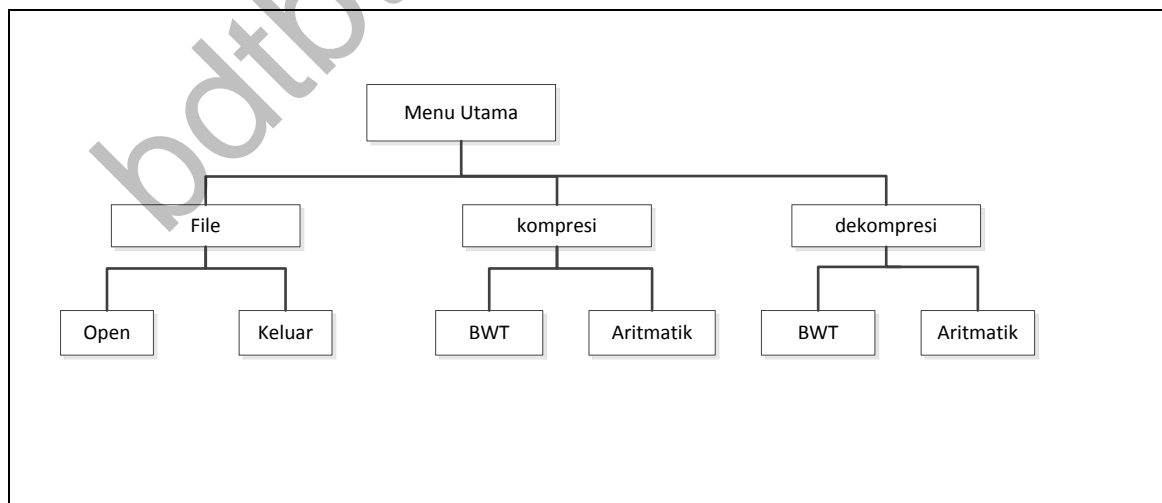
#### 3.1 Perancangan perangkat lunak

Pembuatan Aplikasi yang akan di gunakan adalah menggunakan perangkat lunak IDE delphi XE. Aplikasi yang akan dibuat, berupa perangkat lunak untuk kompresi dan dekompresi, yang menggunakan algoritma Burrow wheeler dan pengkodean Aritmatik. Tujuan pembuatan perngaktat lunak ini adalah untuk menghasilkan sebuah aplikasi kompresi maupun dekompresi sebagai alat bantu dalam analisis.

Perancangan program aplikasi ini terdiri dari beberapa menu yang di desain dalam beberapa item program yaitu :

1. menu file adalah menu untuk membuka dan memilih file yang akan diproses oleh perangkat lunak ini.
2. Menu kompresi adalah menu untuk mengkompresi data yang akan diproses dengan pilihan dua teknik kompresi
3. Menu dekompresi adalah menu untuk mengembalikan data yang telah dikompresi ke dalam bentuk data aslinya
4. Menu keluar adalah menu yang digunakan untuk keluar dari perangkat lunak ini.

Struktur program dari hasil perancangan tersebut bisa dilihat pada gambar 3.1



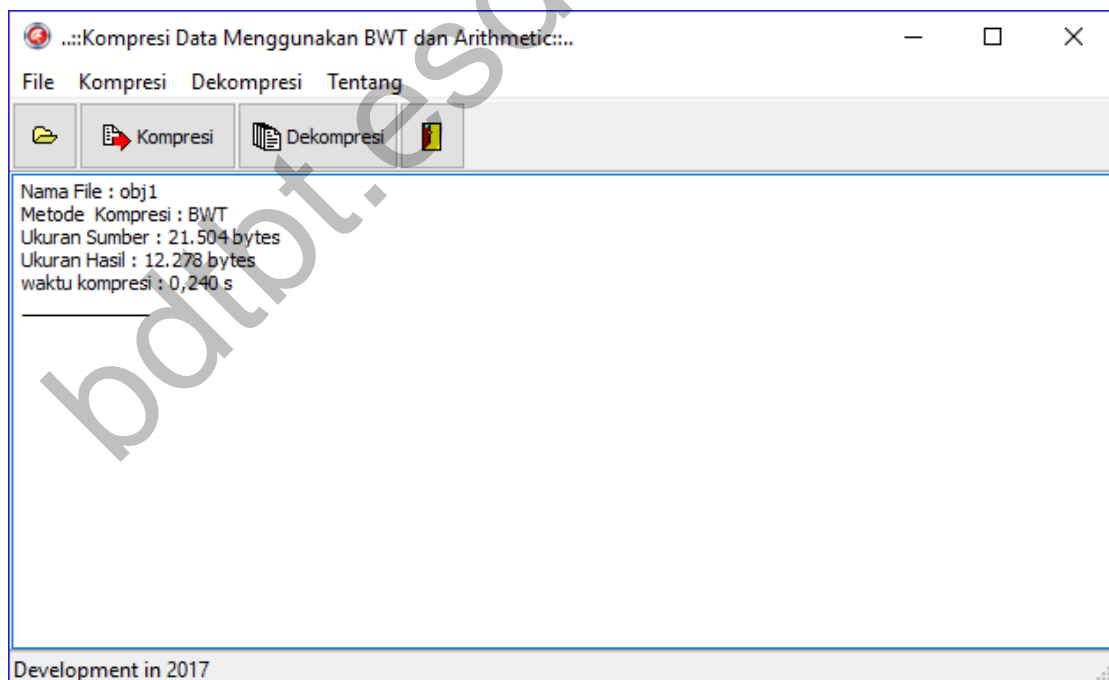
Gambar 3.1 struktur program

#### 3.2 Implementasi

Dalam menganalisis hasil yang akan di dapat penulis menggnkan file standar Calgary Corpus. File calgary corpus merupakan kumpulan beberapa file yang biasa digunakan untuk alat bantu pengujian performansi terhadap beberpa aplikasi pengkompresi data text. Pertama

kali diperkenalkan oleh T.C Bell, J.G. Cleary, dan Witten pada tahun 1989. Pada calgary corpus dipresentasikan sembilan tipe data yang berbeda, dan untuk membuktikan kekonsistensian performansi suatu tipe file. Beberapa tipe dalam calgary corpus yaitu : book1, book2, paper1, paper2, paper3, paper4, paper5 dan paper6 adalah jenis file teks normal berbahasa inggris. Teks yang tidak normal diwakili oleh file new dan bib. Selain file-file ASCII terdapat juga file Non ASCII yaitu obj1 dan obj2, yang merupakan file kode tereksekusi, obj1 untuk VAX dan obj2 untuk Apple Macintosh. File gambar yang memiliki dua warna yaitu hitam dan putih ialah file pic. File progc, progl dan progp yang merupakan file kode sumber, progc berisi kode sumber dalam bahasa C, progl berisi kode sumber dalam bahasa LISP, progp berisi kode sumber dalam bahasa pascal. Data geofisical diwakili dengan file bernama geo.

Hasil yang didapat pada perancangan perangkat lunak dapat dilihat pada gambar 2.1. perangkat lunak tersebut digunakan untuk implementasi analisis proses kompresi data dengan teknik lossless compression.



**Gambar 2.1 Hasil Program Aplikasi kompresi dan dekompresi**

Pengujian yang akan digunakan ialah berupa rasio serta kecepatan kompresi. Analisis terhadap ukuran hasil nilai parameter tersebut akan dilakukan terhadap algoritma yang digunakan serta karakteristik file yang di proses. Untuk mengukur rasio digunakan rumus sebagai berikut yaitu :



$$\text{Rasio} = (\text{ukuran File Terkompresi} / \text{Ukuran File Asli}) * 100\%$$

Setelah dilakukan pengujian terhadap beberapa file Calgary corpus maka didapat hasil yang dapat dilihat pada tabel 1.1.

Nama File	Ukuran Asli (bytes)	BWT		Arithmetic	
		Ukuran (Bytes)	Rasio (%)	Ukuran (Bytes)	Rasio (%)
bib	111261	38245	34,37	72932	65,55
book1	768771	269351	35,04	437038	56,85
book2	610856	199471	32,65	361971	59,26
geo	102400	58434	57,06	72828	71,12
news	377109	149117	39,54	242175	64,22
obj1	21504	12278	57,10	15338	71,33
obj2	246814	102944	41,71	186239	75,46
paper1	53161	23044	43,35	33210	62,47
paper2	82199	33001	40,15	48016	58,41
paper3	46526	20801	44,71	27803	59,76
paper4	13286	6947	52,29	8431	63,46
paper5	11954	6500	54,38	7896	66,05
paper6	38105	17290	45,37	23840	62,56
pic	513216	50263	9,79	68736	13,39
progc	39611	17526	44,25	25984	65,60
progl	71646	22676	31,65	42632	59,50
progp	49379	17012	34,45	30257	61,28

Tabel 1.1 Hasil proses kompresi

Hasil analisis terhadap rasio kompresi dari algoritma BWT dan algoritma Aritmatik penulis lakukan yaitu :

- Pada table 1.1 hasil kompresi dapat diketahui bahwa rasio kompresi rata-rata untuk algoritma BWT adalah 40,05% dan untuk algoritma aritmatik 60,96 %, sehingga terlihat bahwa BWT lebih unggul dari rata-rata rasio kompresi data. Nilai performansi rasio kompresi yang diperoleh suatu pengkodean bergantung pada probabilitas yang dibangun..
- Kecepatan kompresi rata-rata untuk algoritma BWT adalah 4202 bytes/s dan untuk algoritma aritmatik adalah 5905 bytes/s. Hal ini berarti algoritma BWT memberikan tambahan waktu dibanding algoritma aritmatik.
- Dari data hasil tabel 1.1 menunjukkan bahwa proses transformasi kompresi pada BWT dapat membuat rasio kompresi yang lebih baik dari algoritma kompresi aritmatik tetapi harus mengorbankan kecepatan.



#### 4. Kesimpulan

Algoritma BWT mampu menghasilkan proses kompresi data yang baik dibandingkan dengan pengkodean aritmatik. Walaupun terjadi penurunan kecepatan, namun waktu kompresi keseluruhan perbandingannya tidak terlalu signifikan. Performansi kompresi sangat bergantung pada sumber data masukan. File dengan sedikit ragam simbol akan menghasilkan rasio kompresi lebih tinggi dibandingkan dengan yang memiliki banyak ragam simbol. Performansi dapat diperbaiki dengan perbaikan pada perangkat lunak dan perangkat keras yang lebih baik lagi.

bdtbt.esdm.go.id





## DAFTAR PUSTAKA

- [1]. Burrows, M and Wheeler , D,J . *A Blok Sorting Lossless Data Compression Algorithm* , Digital System Research Center Report, 124, 1994
- [2]. Nelson, Mark., *Data Compression with the Burrows Wheelers Transform* , Dr.Dobb's Journal, 1996
- [3]. Nelson, Mark, *The Data Compression Book*, M&T Publishing, 1996
- [4]. Jogianto. *Pengenalan Komputer*. Yogyakarta : PT. Andi Offset, 1989
- [5]. Kadir, Abdul. *Dasar Pemograman Delphi 5.0 Jilid 2*. Yogyakarta : PT, Andi Offset, 2001

batbt.esdm.go.id