

IMPLEMENTASI KOMPRESI DATA PADA JARINGAN KOMPUTER MENGUNAKAN ALGORITMA ZLIB

Hapid Rahmat¹, Dhami Johar Damiri², Ate Susanto³

Jurnal Algoritma
Sekolah Tinggi Teknologi Garut
Jl. Mayor Syamsu No.2, Tarogong Kidul, Garut
Email: jurnal@sttgarut.ac.id

¹abdul_hapidrahmat@yahoo.co.id

²djdamiri@yahoo.co.id

³atesapooh@gmail.com

Abstrak - Pengiriman data atau file melalui media jaringan tentu memerlukan *bandwidth* yang besar apabila ukuran file yang akan dikirim juga besar, sehingga proses pengiriman data yang berkapasitas besar itu akan mengakibatkan jaringan yang ada menjadi sibuk dan membuat pengiriman file memakan waktu yang cukup lama, maka dari itu diperlukan sebuah aplikasi yang berfungsi untuk mengkompresi ukuran file, sehingga pada saat pengiriman data dapat menghemat penggunaan waktu transfer data. Salah satu algoritma untuk mengkompresi data adalah algoritma ZLIB. Pemakaian Algoritma ZLIB ini karena sifatnya yang *Loseless*, dimana teknik kompresi data hasil kompresi dapat didekompres lagi dan hasilnya sama seperti data sebelum proses kompresi, sehingga proses transfer file akan menjadi lebih maksimal dan mengurangi waktu transfer file.

Kata Kunci - Kompresi Data, Algoritma Zlib

I. Pendahuluan

Data mempunyai peranan yang sangat penting bagi setiap orang yang setiap harinya menggunakan media komputer. Sehingga banyak *file* atau data yang disebar melalui media jaringan ataupun media lainnya. Akan tetapi pada saat pengiriman data atau *file* melalui media jaringan tentu memerlukan *bandwidth* yang besar apabila ukuran *file* yang akan dikirim juga besar, begitu pula dengan jaringan lokal yang memerlukan banyak ruang untuk kelancaran komunikasi data, sehingga proses pengiriman data yang berkapasitas besar itu akan mengakibatkan jaringan menjadi sibuk sehingga pengiriman data atau *file* memakan waktu yang lama.

Sehubungan dengan permasalahan di atas maka diperlukan sebuah aplikasi yang berfungsi untuk mengecilkan ukuran *file*, sehingga pada saat pengiriman data dapat menghemat penggunaan waktu transfer data. Salah satu algoritma untuk mengkompresi data adalah algoritma ZLIB.

Penggunaan Algoritma ZLIB bersifat *Loseless*, dimana hasil kompresi dapat didekompres lagi dan hasilnya tepat sama seperti data atau *file* sebelum dikompresi, sehingga proses pengiriman *file* akan mereduksi waktu *transfer file*.

Secara spesifik, kompresi data atau *file* dengan menggunakan Algoritma ZLIB bertujuan untuk mereduksi tempat (*space*) penyimpanan data atau *file* dan mereduksi waktu untuk mentransmisikan data atau *file* yang memiliki kapasitas besar.

II. Tinjauan Pustaka

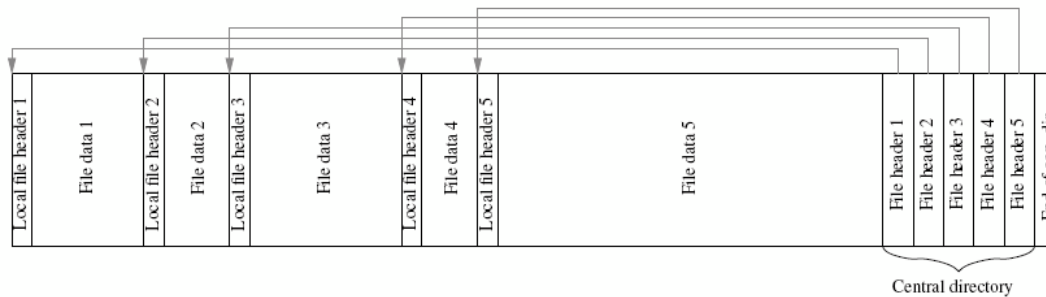
A. Kompresi Data

Kompresi data dalam bidang ilmu komputer, ilmu pengetahuan dan seni adalah sebuah penyajian informasi ke dalam bentuk yang lebih sederhana [1]. Kompresi data dapat di artikan juga sebagai proses yang dapat mengubah sebuah aliran data masukan (sumber atau data asli) ke dalam aliran data yang lain (keluaran atau data yang dimampatkan) yang memiliki ukuran yang lebih kecil.[2]

B. Algoritma Kompresi ZLIB

Algoritma ZLIB merupakan turunan dari Agoritma kompresi *Deflate*. Algortima ini diciptakan oleh Jean-Loup Gailly yang menciptakan proses kompresi data dan Mark Adler yang menciptakan proses dekompresi data. Algoritma Zlib melakukan kompresi dengan mengkompresikan data yang terdiri dari serangkaian blok, sesuai dengan blok inputan data tersebut. Setiap blok pada data tersebut dikompresi dengan menggunakan algoritma kompresi data *Deflate* sebagai kompressor yang merupakan variasi dari Algoritma LZ77 dikombinasikan dengan *Huffman Coding*. [3]

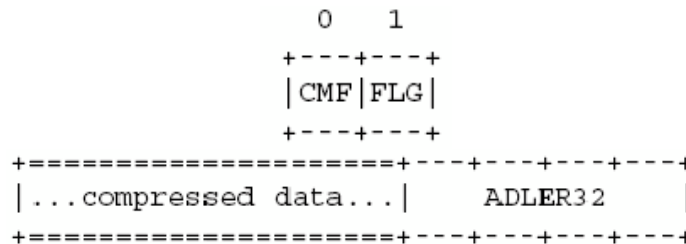
Data yang dikompresi oleh algoritma ZLIB akan diberikan pembungkus berupa data *header* setelah data tersebut dikompresi oleh algoritma *Deflate*. *Header file* yang diberikan pada data yang telah dikompresi tersebut merupakan bagian dari Algoritma ZLIB. Adapun struktur *file* dari agoritma ZLIB beserta *header file* yang telah diberikan pada data yang telah dikompresi ditunjukkan pada gambar 1.



Gambar 1. Struktur *File* ZLIB

C. Format Data ZLIB

Struktur format data Algoritma ZLIB memiliki struktur seperti pada gambar 2 di atas dimana format data yang diletakkan setelah data kompresi merupakan *header* dari data Algoritma ZLIB :



Gambar 2 Struktur format data ZLIB

Dari format data Zlib tersebut, setiap data yang mungkin muncul setelah ADLER32 bukan bagian dari Algoritma ZLIB.

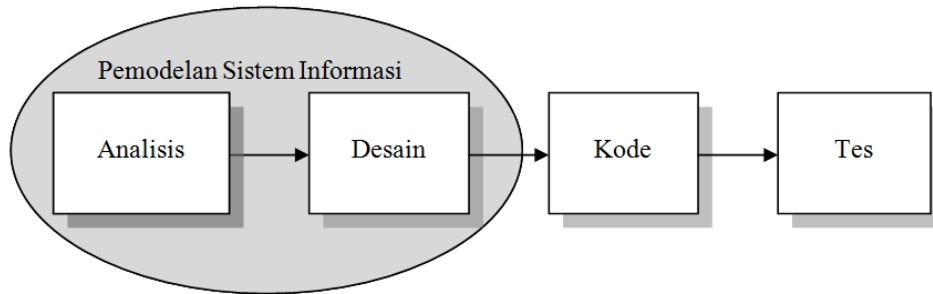
- *CMF* (*Compression Method and Flags*)

Byte ini dibagi menjadi metode kompresi 4-bit dan *field* informasi yang bergantung pada metode kompresi.

- *CM (Compression Method)*
Mengidentifikasi metode kompresi yang digunakan pada *file* tersebut. Contohnya CM=8 menunjukkan bahwa metode kompresi yang digunakan adalah metode “*Deflate*”.
- *CINFO (Compression info)*
Untuk CM=8, CINFO adalah algoritma dasar yang digunakan yaitu algoritma LZ77.
FLG (FLaGs)
Flag pada sebuah *byte* dibagi antara lain sebagai berikut :
Bit 0 – 4 FCHECK (untuk memeriksa bit CMF dan FLG)
Bit 5 FDICT
Bit 6 – 7 FLEVEL (Tingkat kompresi)
- *FDICT (Preset Dictionary)*
Jika FDICT diatur, *identifier* dari kamus DICT diletakkan setelah *byte* FLG. Dimana kamus dari data tersebut adalah urutan *byte* awal pada kompresor yang tidak menghasilkan *output* terkompresi. DICT adalah urutan *byte checksum Adler-32*. Dekompresor ini dapat digunakan untuk menentukan kamus yang telah digunakan oleh kompresor.
- *FLEVEL (Level kompresi)*
Flag ini digunakan untuk menentukan metode kompresi tertentu. Contohnya metode “*deflate*” (CM = 8) flag ini ditentukan sebagai berikut.
0 – kompresor menggunakan Algoritma tercepat
1 – kompresor menggunakan Algoritma cepat
2 – kompresor menggunakan Algoritma default
3 – kompresor menggunakan Algoritma kompresi maksimum, algoritma paling lambat.
Informasi didalam FLEVEL tidak diperlukan untuk melakukan proses dekompresi, melainkan untuk mengindikasikan jika proses kompresi diulangi.
- *Compressed Data*
Bagian ini merupakan *file* yang telah diberikan proses kompresi. Contohnya algoritma kompresi ZLIB menggunakan algoritma *deflate* sebagai algoritma kompresor nya.
- *ADLER32*
Ini berisi nilai *checksum* dari data terkompresi (tidak termasuk kamus data) yang dihitung menurut algoritma Adler-32. Algoritma ini adalah ekstensi 32-bit dan perbaikan dari algoritma *Fletcher*, yang digunakan dalam ITU-T X.224/standar ISO8073. Adler-32 terdiri dari jumlah dua akumulasi per *byte*: s1 adalah jumlah dari semua *byte*, s2 adalah jumlah dari semua nilai s1. Jumlah dari keduanya terdiri dari 65.521 modul. s1 diinisialisasi dengan 1, s2 diinisialisasi dengan 0. Algoritma Adler-32 memeriksa jumlah yang disimpan sebagai $s2*65.536 + s1$ dalam *most-significant-byte* yang pertama. [4]

III. Metode Penelitian

Tahapan pembuatan perangkat lunak ini terdiri dari sederetan kegiatan yang dapat dikelompokkan menjadi beberapa tahapan, yang membantu kita dalam pengembangan sistem. Metodologi pengembangan perangkat lunak yang akan digunakan adalah SDLC (*System Development Life Cycle*) yang memiliki beberapa model dalam penerapan tahapannya diantaranya model Sekuensial Linier atau alur hidup klasik (*classic life cycle*) seperti pada gambar 3.



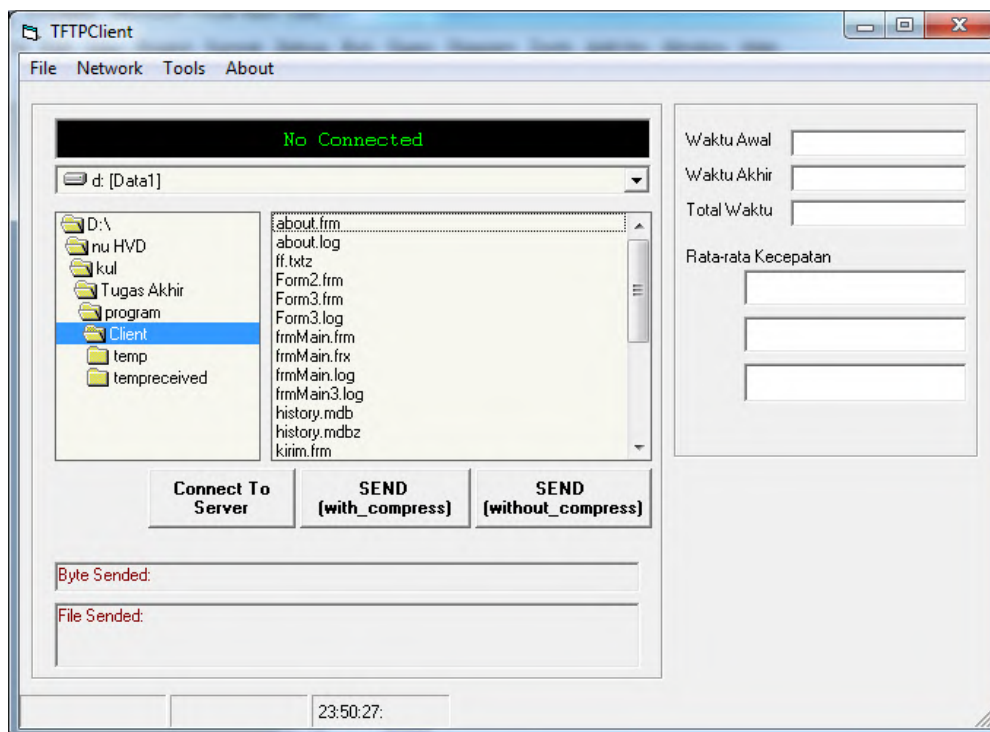
Gambar 3 Sekuensial Linier [5]

IV. Hasil dan Pembahasan

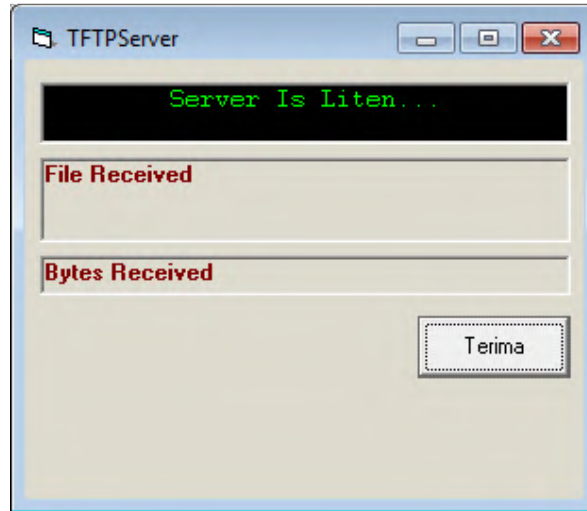
Kompresi data dengan menggunakan algoritma ZLIB ini dibangun dengan menggunakan Bahasa Pemrograman *Visual Basic 6.0*. Program kompresi data ini terdiri dari beberapa bagian. Bagian kompresi, bagian dekompresi dan bagian *client server* yang dapat melakukan kompresi dan dekompresi pada saat data dikirimkan. Berikut merupakan gambar tampilan setiap bagian dari aplikasi yang telah dibuat.

A. Tampilan Utama

Tampilan utama merupakan tampilan yang pertama kali muncul ketika program tersebut dijalankan. Tampilan utama ditunjukkan pada gambar 4. Berikut :



Gambar 5 Tampilan Utama Aplikasi Kompresi ZLIB untuk mengirim file



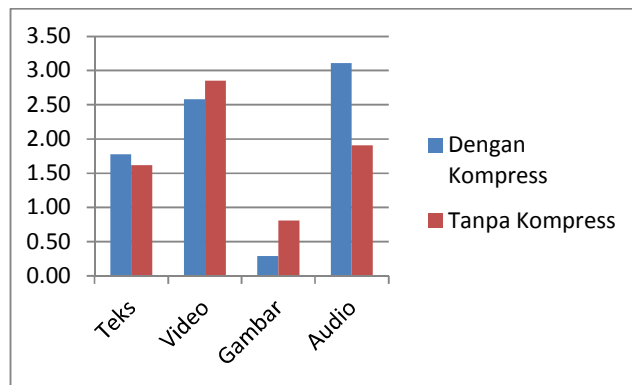
Gambar 6 Tampilan Utama Aplikasi Kompresi ZLIB untuk komputer penerima *file*

B. Hasil Pengujian

Dari hasil berbagai pengujian setiap jenis dan size *file* yang berbeda dengan menggunakan kompresi algoritma ZLIB maka dapat ditentukan rata-rata waktu dan selisih untuk pengujian masing-masing jenis *file* seperti yang terlihat pada tabel 1 berikut:

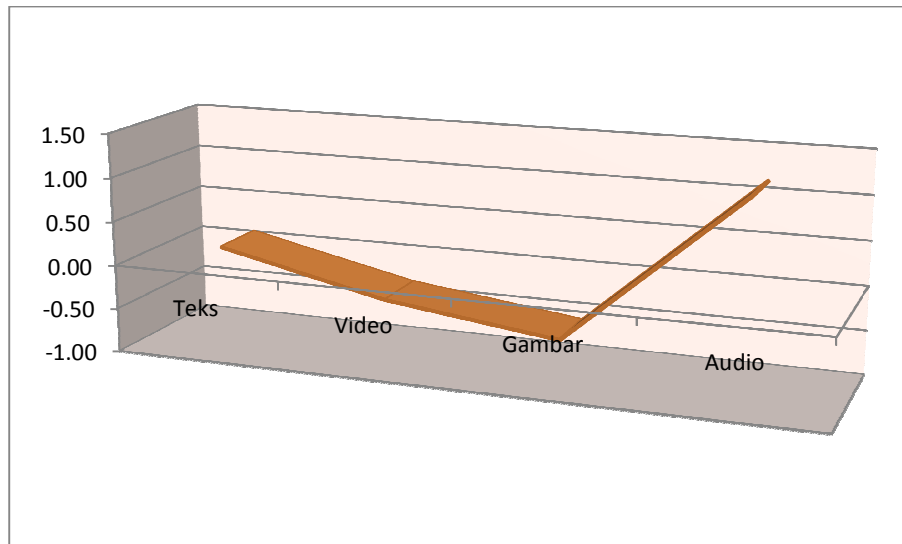
Tabel 1 rata-rata dan selisih waktu pengiriman *file* dengan menggunakan teknik kompresi algoritma ZLIB dan tanpa kompresi

| Tipe File | Rata-rata Waktu Pengiriman (ms) | | Selisih |
|-----------|---------------------------------|---------------|---------|
| | Dengan Kompres | Tanpa Kompres | |
| Teks | 1,77 | 1,62 | 0,16 |
| Video | 2,58 | 2,85 | -0,27 |
| Gambar | 0,29 | 0,81 | -0,52 |
| Audio | 3,11 | 1,91 | 1,20 |



Gambar 7 grafik rata-rata dan selisih waktu pengiriman *file* dengan menggunakan teknik kompresi algoritma ZLIB dan tanpa kompresi

Berdasarkan perbedaan antara waktu pengiriman *file* antara yang menggunakan kompresi *file* ZLIB dan yang tidak menggunakan kompresi *file* ZLIB dapat digambarkan perbedaan untuk tiap-tipe *file* pada gambar 8 berikut:



Gambar 8 hasil kompresi data dari berbagai jenis dan *file size* yang berbeda

Dari gambar 4.12 di atas menunjukkan hasil kompresi data yang paling baik ditujukan pada tipe *file* gambar yang selisih nilainya sebesar 0,72 ms.

V. Kesimpulan

Setelah melakukan analisis dan pengujian pada sistem yang dibangun, maka dapat diperoleh kesimpulan sebagai berikut :

1. Rasio hasil kompresi yang paling signifikan menggunakan algoritma ZLIB adalah tipe file gambar dengan ekstensi .bmp dengan rata-rata tingkat rasio hasil kompresi sebesar 1 : 0,04.
2. Perbedaan waktu pengiriman yang paling signifikan diperoleh paling cepat pada tipe file gambar dengan rata-rata waktu pengiriman dengan menggunakan kompresi data ZLIB, rata-rata waktu yang dibutuhkan adalah 0,06 ms, dan yang tidak dilakukan kompresi data sebesar rata-rata waktu yang dibutuhkan adalah 0,04 ms.
3. Algoritma ZLIB baik digunakan untuk tipe file yang ber-ekstensi .bmp karena tipe file ini merupakan tipe gambar yang belum terkompresi, berbeda halnya dengan tipe .jpg dan .png yang merupakan tipe file yang sudah terkompresi.

VI. Daftar Pustaka

- [1] Mengyi Pu, Ida. "Fundamental Data Compression", Linacre House, Jordan Hill, Oxford. 2006.
- [2] Salomon, David, "Data Compression" Edisi 4th, Computer Science Department, California State " University, Northridge, Springer-Verlag, London, 2007.
- [3] ZLIB, www.zlib.net, 02 Juni 2012 (www.zlib.net, diakses tanggal 02 Juni 2012).
- [4] Deutsch, L. Peter, and Gailly, Jean-Loup, ZLIB "Compressed Data Format Specification version 3.3" Network Working Group.
- [5] Pressman, Roger S., " Rekayasa Perangkat Lunak", Mc Graw-Hill Companies, 1997.