

Aplikasi Kompresi Untuk Pengiriman Data Menggunakan Metode LZW (*Lemple Ziv Welch*)

Imam Marzuki

Program Studi Teknik Elektro, Fakultas Teknik, Universitas Panca Marga Probolinggo
Jl. Yos Sudarso No. 107, Pabean, Dringu, Probolinggo 67271
Email : imamarzuki32@gmail.com

Terima Naskah : 7 September 2017

Terima Revisi : 29 September 2017

ABSTRAK

Dalam komunikasi data, pesan yang dikirim seringkali menjumpai berbagai macam problem dalam proses pengiriman, seperti transfer data yang memakan waktu cukup lama dalam prosesnya. padahal orang menginginkan data dengan waktu yang singkat. masalah tersebut dapat diatasi dengan mengkodekan pesan atau isi arsip sesingkat mungkin, sehingga waktu pengiriman pesan atau data juga relatif cepat, dan ruang penyimpanan yang dibutuhkan juga sedikit. Cara pengkodean seperti ini disebut pemampatan atau kompresi data. Dari sekian banyak algoritma kompresi, penulis tertarik untuk mencoba menggunakan algoritma LZW (*Lemple Ziv Welch*) karena algoritma ini merupakan salah satu algoritma paling populer untuk penyimpanan *lossless*. Hasil Implementasi algoritma LZW pada pemrograman java dan My SQL berhasil melakukan pengkompresian dan transfer data dengan baik, data yang sudah terkompresi menjadi lebih kecil.

Kata kunci : Kompresi, Dekompresi, LZW (*Lemple Ziv Welch*)

ABSTRACT

In data communication, messages sent often encounter various kinds of problems in the sending process, such as data transfer that takes a long time in the process. even though people want data with a short time. the problem can be overcome by encoding the message or the contents of the archive as short as possible, so that the time of sending messages or data is also relatively fast, and the storage space needed is also small. This way of coding is called compression or data compression. Of the many compression algorithms, the authors are interested in trying to use the LZW algorithm (Lemple Ziv Welch) because this algorithm is one of the most popular algorithms for lossless storage. Results The implementation of the LZW algorithm in java programming and My SQL successfully compresses and transfers data properly, the compressed data becomes smaller.

Keywords : Compression, Decompression, LZW (*Lemple Ziv Welch*)

PENDAHULUAN

Dalam komunikasi data, pesan yang dikirim seringkali menjumpai berbagai macam problem dalam proses pengiriman, seperti transfer data yang memakan waktu cukup lama dalam prosesnya. padahal orang menginginkan data dengan waktu yang singkat. masalah tersebut dapat diatasi dengan mengkodekan pesan atau isi arsip sesingkat mungkin, sehingga waktu pengiriman pesan atau data juga relatif cepat, dan ruang penyimpanan yang dibutuhkan juga sedikit.

Perkembangan teknologi sangat cepat dan pesat, hal ini yang menyebabkan munculnya kemajuan teknologi. Dengan demikian, teknologi

telah menjadi bagian penting dari berbagai bidang kehidupan manusia, khususnya dibidang pengkompresian. Sehingga dapat mengkompresi data ke ukuran yang lebih kecil dan memungkinkan pengiriman data lebih cepat. Dengan perkembangan teknologi ini, banyak aplikasi-aplikasi yang tersedia. Pada saat ini teknologi kompresi data berkembang pesat dikehidupan masyarakat.

Hal ini tentu saja dapat menimbulkan resiko, jika data tidak terkompresi dengan baik. Maka akan mengalami kerugian antara lain: ruang penyimpanan pada hardisk akan cepat penuh dan pengiriman data menjadi lebih lama. Yang mana jika hal tersebut sampai terjadi, kemungkinan

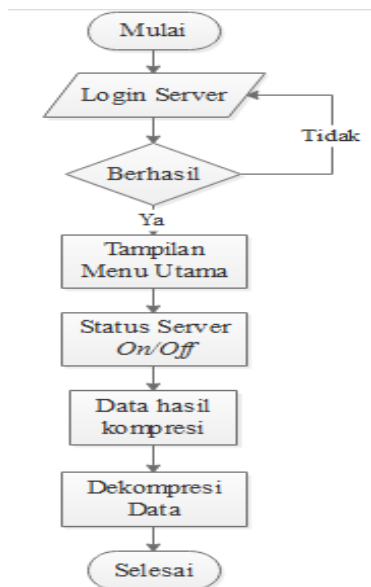
besar akan merugikan penggunanya. Maka diperlukannya teknologi kompresi data agar data tersebut tidak banyak memerlukan ruang pada hardisk dan pengiriman datanya lebih cepat.

Java adalah bahasa pemrograman yang dapat dijalankan diberbagai komputer termasuk telepon genggam. Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi. Aplikasi java mampu berjalan dibeberapa platform sistem operasi yang berbeda.

Dengan adanya kompresi data diharapkan dapat menghemat biaya serta waktu yang dikeluarkan guna menambah fasilitas media penyimpanan data pada komputer serta mempercepat proses transfer data. Maka berdasarkan latar belakang yang telah dikemukakan di atas, judul yang diambil dalam penelitian ini adalah Aplikasi Kompresi Untuk Pengiriman Data Menggunakan Metode LZW (*Lemple Ziv Welch*), yang menggunakan bahasa pemrograman *java* dan pengolah *database MySql*. Penelitian ini diharapkan dapat memberikan solusi yang lebih baik dalam pengiriman data.

METODE

Flowchart kompresi data pada server

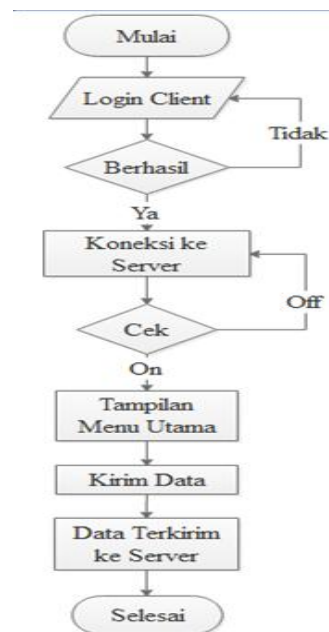


Gambar 1. Flowchart Kompresi Data pada Server

Flowchart pada gambar 1 menjelaskan bahwa untuk melakukan sebuah aplikasi kompresi data

pada server, yaitu dengan memulai pada suatu aplikasi kompresi tersebut. Kemudian aplikasi menampilkan tampilan *form login* pada server, jika *User* (pengguna) melakukan *login* salah memasukkan *username* atau *password* maka aplikasi akan melakukan login kembali sampai *username* dan *password* benar, jika *login* berhasil dilakukan maka akan muncul menu utama pada aplikasi server, selanjutnya *user* diserver melakukan koneksi terlebih dahulu agar client bisa terhubung dengan server, setelah itu kita akan mendekompresi data yang telah dikirim oleh client dan selesai.

Flowchart kompresi data pada client



Gambar 2. Flowchart Kompresi Data pada Client

Flowchart pada gambar 2 menjelaskan alur dari aplikasi kompresi data pada client. Pertama kita mulai untuk menjalankan aplikasi tersebut, selanjutnya akan menampilkan *form login* pada client, jika *user* (pengguna) melakukan login salah memasukkan *username* atau *password*, maka aplikasi akan melakukan login kembali sampai *username* dan *password* benar, kemudian cek koneksi dari server jika koneksi atau status server masih *Off* client tidak akan bisa masuk pada menu utama di client, begitu sebaliknya jika status server *On* client bisa masuk pada menu utama di client, setelah itu akan menampilkan menu utama pada client, kemudian kita pilih data yang akan kita

kirim, setelah itu kirim data, kemudian aplikasi akan menjalankan proses kompresi data menggunakan algoritma LZW dan data akan terkirim ke server setelah itu selesai.

Uji Coba Metode LZW (Lemple Ziv Welch)

Sebagai contoh karakter “IMAM MARZUKI” akan di kompresi dengan menggunakan metode LZW (Lemple Ziv Welch), dimana kata IMAM MARZUKI tersebut memiliki 12 karakter atau mempunyai ukuran sebesar 12 byte kemudian kita buat tabel metode LZW (Lemple Ziv Welch) untuk mempermudah perhitungannya.

Tabel 1. Karakter Sebelum Dikompresi

I	M	A	M		M	A	R	Z	U	K	I
73	77	65	77	32	77	65	82	90	85	75	73

Tabel 2. Hasil atau Output Kompresi

I	M	A	M			R	Z	U	K	I
73	77	65	77	32	257	82	90	85	75	73

Jadi output yang di hasilkan berupa kode ASCII 73 77 65 77 32 257 82 90 85 75 73, yang awalnya IMAM MARZUKI memiliki 12 karakter atau ukuran 12 byte setelah di kompresi ke metode LZW mengalami pengurangan karakter menjadi 11 karakter atau pengurangan ukuran menjadi 11 byte.

Uji coba dekompresi LZW dari karakter IMAM MARZUKI yang telah di kompresi tadi sudah terkompresi ke dalam kode ASCII kita akan kembalikan file tersebut ke ukuran semula menjadi 12 karakter.

Tabel 3. Uji Coba Metode Dekompresi LZW

Input	S	K	S+K	DIC		Output
				C	C	
73	I	null	Null	null	null	I
77	I	M	IM	256	IM	M
65	M	A	MA	257	MA	A
77	A	M	AM	258	AM	M
32	M	(spasi)	M(spasi)	259	M(spasi)	(spasi)
257	(spasi)	MA	(spasi)MA	260	(spasi)MA	MA
82	MA	R	MAR	261	MAR	R
90	R	Z	RZ	262	RZ	Z
85	Z	U	ZU	263	ZU	U
75	U	K	UK	264	UK	K
73	K	I	KI	265	KI	I

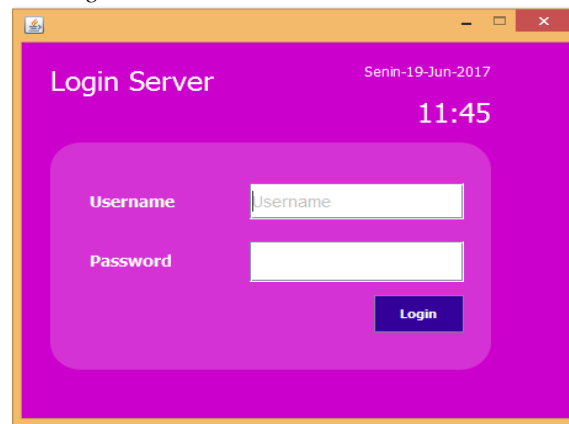
Untuk melakukan dekompresi, Jika ada masukan data karakter S, maka output kode adalah S, selanjutnya nilai K diisi dengan karakter S. Ada satu variable tambahan yaitu entry yang digunakan untuk mengambil satu karakter dari S untuk dijadikan Output Pertama, selanjutnya nilai K adalah Output berikutnya sampai karakter terakhir.

HASIL DAN PEMBAHASAN

Implementasi Antarmuka

Berikut tampilan implementasi aplikasi kompresi data ini:

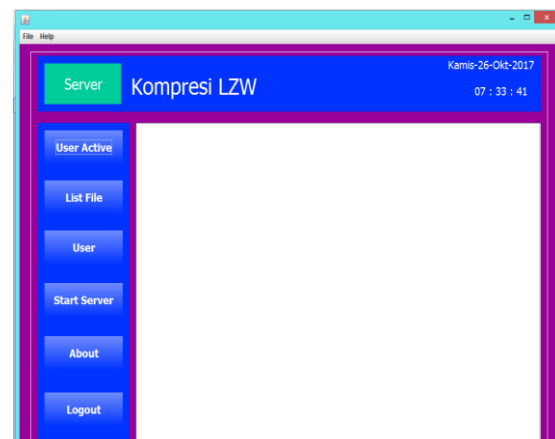
a. Login



Gambar 3. Tampilan Login Server

Pada saat aplikasi dijalankan, pengguna atau user akan ditampilkan form login pada server terlebih dahulu. Login ini bertujuan untuk masuk ke halaman menu utama.

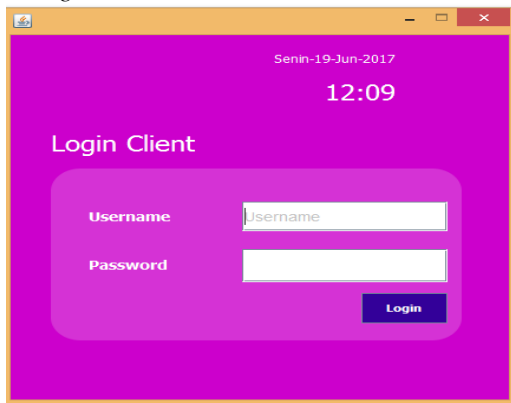
b. Tampilan Menu Utama Server



Gambar 4. Tampilan Menu Utama Server (sumber: data diolah oleh penulis)

Pada menu utama ini terdapat beberapa tombol yang memiliki fungsi masing-masing yaitu, tombol *list* file berfungsi untuk menampilkan file yang sudah dikompresi, tombol *Start Server* berfungsi untuk mengkoneksikan ke aplikasi *client*, tombol *user* berfungsi untuk membuka *form* data *user* untuk mendaftarkan menjadi admin ataupun *client*, tombol *About* berfungsi untuk menampilkan profil pribadi, dan tombol *Logout* berfungsi untuk keluar dari aplikasi *server* tersebut.

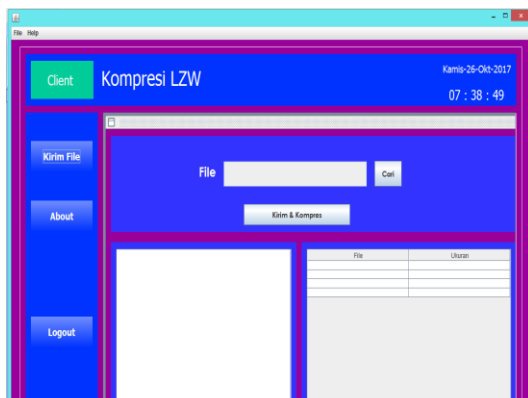
c. Login Client



Gambar 5. Login Client
(sumber: data diolah oleh penulis)

Pada form *login* di *client* ini, *user* diharapkan untuk *login* kembali dengan *username* dan *password* yang berbeda dengan *username* dan *password* yang dimiliki *server* jika *user* menggunakan milik *server* maka akan muncul pemberitahuan “hak akses salah” dan ketika *client* ingin masuk ke halaman utama, tetapi pada *server* belum hidup maka akan muncul pemberitahuan “hidupkan *server* terlebih dahulu”.

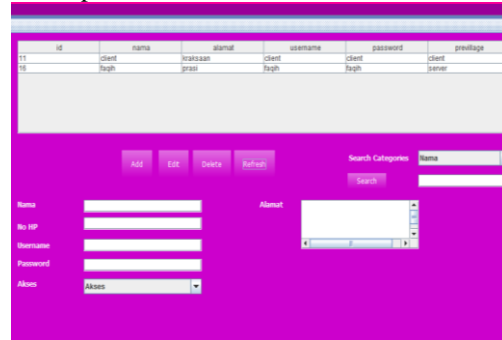
d. Tampilan Menu Utama Client



Gambar 6. Tampilan Menu Utama Client
(sumber: data diolah oleh penulis)

Pada tampilan menu utama *client* ini, terdapat beberapa tombol yang memiliki fungsi masing-masing. Tombol *Cari* berfungsi untuk mencari file yang akan dikirim, tombol *Kirim* berfungsi untuk mengirim file sekaligus berfungsi untuk melakukan proses kompresi data yang akan dikirim tadi, tombol *About* berfungsi untuk menampilkan profil pribadi pembuat aplikasi ini, dan tombol *logout* berfungsi untuk keluar dari aplikasi *client*.

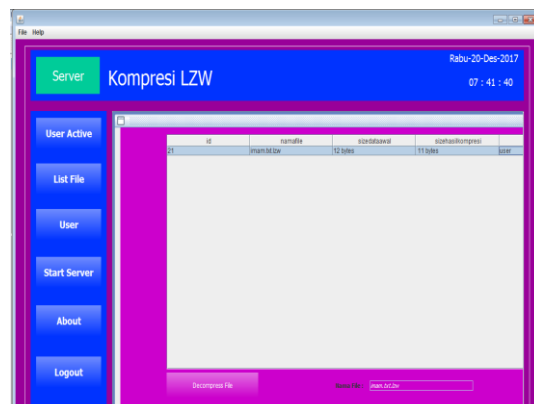
e. Tampilan User



Gambar 7. Tampilan User
(sumber: data diolah oleh penulis)

Di tampilan data *user* berfungsi untuk mendaftarkan diri *user* agar menjadi admin atau *client*. Terdapat beberapa tombol yang ada didalam tampilan *user* yang memiliki fungsi masing-masing, tombol *save* berfungsi untuk menyimpan data *user* yang sudah diinput, tombol *edit* berfungsi untuk merubah data *user*, tombol *delete* berfungsi untuk menghapus data *user*, Tombol *Refresh* berfungsi untuk mengembalikan tampilan *user* dalam keadaan semula dan Tombol *Search* berfungsi untuk mencari data *user* yang ingin kita cari.

f. Tampilan List File



Gambar 8. Tampilan List File

Tampilan *list* file ini berfungsi untuk menampilkan file yang sudah di kompresi, dan terdapat tombol dekompresi file yang Berfungsi untuk menampilkan file yang sudah didekompresi.
g. Tampilan *About*



Gambar 9. Tampilan *About*

Pada tampilan form *About* ini berfungsi untuk menampilkan data profil pribadi pembuat program.

Pengujian terhadap data teks txt

Pengujian ini bertujuan untuk mengetahui sebuah hasil yang diterapkan dalam sebuah pemrograman yang telah dibuat.

Tabel 4. Data ≥ 1000 bytes

id	namafile	sizedataawal	sizehasilkompresi	Presentase Kompresi
1	coba1.txt.lzw	1201 bytes	600 bytes	601 bytes
2	coba2.txt.lzw	1316 bytes	658 bytes	658 bytes
3	coba3.txt.lzw	1198 bytes	530 bytes	668 bytes
4	coba4.txt.lzw	1201 bytes	605 bytes	596 bytes
5	coba5.txt.lzw	1638 bytes	663 bytes	975 bytes

Tabel diatas menjelaskan hasil kompresi data terhadap beberapa data teks txt yang ukuran ≥ 1000 bytes dimana ukuran file awal lebih besar dari pada file yang sudah dikompresi, sehingga pengujian diatas berhasil sesuai dengan yang diharapkan. Dari percobaan diatas tidak terlalu terlihat persentasenya yang rata-rata bisa di kompres sekitar ≥ 600 bytes.

Percobaan kedua disini akan dilakukan dengan data yang lebih besar ≥ 4000 bytes.

Tabel 5. Data ≥ 4000 bytes

id	namafile	sizedataawal	sizehasilkompresi	Presentase Kompresi
1	bab 1.txt.lzw	4000 bytes	1592 bytes	2408 bytes
2	bab 2.txt.lzw	4960 bytes	1838 bytes	3122 bytes
3	bab 3.txt.lzw	5232 bytes	1941 bytes	3291 bytes
4	bab 4.txt.lzw	5366 bytes	1898 bytes	3468 bytes

Dari tabel diatas dapat diketahui hasil, bahwa presentase ditentukan dengan banyaknya karakter yang sama, dari percobaan \geq bab3 data awal sebesar 5232 bytes hasil kompresinya 1941 bytes presentase kompresi yang di hasilkan sebesar 3291 bytes dan percobaan yang dilakukan terhadap data \geq bab4 yang dimana data awal sebelum dikompresi sebesar 5366 bytes setelah dikompresi hasilnya sebesar 1898 bytes kemudian hasil presentase yang didapat sebesar 3468 bytes dari percobaan antara \geq bab3 dan \geq bab4 dapat disimpulkan bahwa data yang lebih besar bisa saja hasil kompresinya lebih kecil jika dalam sebuah data memiliki banyak karakter yang sama maka presentase kompresi akan lebih besar dan hasil kompresinya lebih kecil.

SIMPULAN

Adapun kesimpulan yang dapat diperoleh dari aplikasi kompresi data ini yaitu:

1. Aplikasi kompresi data berbasis java dengan algoritma LZW (*Lemple Ziv Welch*) dirancang dengan menggunakan Netbeans dan UML.
2. Dari hasil percobaan yang telah dilakukan menunjukkan bahwa aplikasi kompresi data ini bisa mengkompresi data dengan baik, dengan hasil kompresi yang sesuai algoritma LZW dimana data yang lebih besar hasil kompresinya bisa lebih kecil dengan isi data karakter lebih banyak yang sama begitu sebaliknya, jadi hasil tidak ditentukan dengan jumlah bytes melainkan dengan banyaknya susunan karakter yang sama di dalam sebuah data.

DAFTAR PUSTAKA

- [1] Abdullah, Muhammad Maulana. 2009. Kompresi String Menggunakan Algoritma LZW dan Huffman. Bandung
- [2] Dzulhaq, Muhammad Iqbal dan Aan Ahmad Andayani. 2014. Aplikasi Kompresi File Dengan Metode Lempel-Ziv-Welch: STMIK Bina Sarana global
- [3] Neta, Maria Roslin Apriani. 2013 : Perbandingan Algoritma Kompresi Terhadap Objek Citra Menggunakan Java. Yogyakarta: Universitas Atma Jaya Yogyakarta
- [4] Fadil, Muhammad dkk. 2017. Simulasi Kompresi Data Text Menggunakan Algoritma LZW. Bandung
- [5] Wijaya, Aditya dan Suryarini Widodo. Kinerja dan Performa Algoritma Kompresi Loseless

- Terhadap Objek Citra Digital. Depok : Universitas Gunadarma
- [6] Yudianto, M Jafar Noor. 2003-2007 : Jaringan komputer dan pengertiannya, URL : <http://jafaryudianto.blogspot.com/>, diakses pada tanggal 10 november 2016
- [7] Widayati, sri. 2012. Pengertian tramisi data, URL: <http://www.g-excess.com/pengertian-transmisi-data.html>, diakses tanggal 14 november 2016
- [8] Saputra, Didi Indra. 2009. Bahasa Pemrograman Java, URL : <https://didiindra.wordpress.com/tag/pengertian-java/>, diakses tanggal 20 September 2016, 11:05:16
- [9] Hidayat dkk. 2013. Implementasi Algoritma Kompresi LZW Pada Database Server. Bandung: Unikom
- [10] Arjun, Fahmi. 2014. Pengertian Kompresi Dan Dekompresi, URL: <http://poltekharber1eks3.blogspot.co.id/2014/10/pengertian-kompresi-dan-dekompresi.html>, diakses tanggal 13 September 2016, 10:28:29
- [11] Widhiarta, Putu. 2009. Pengantar Kompresi, URL: <http://widhiartha.multiply.com>, diakses tanggal 15 September 2016, 10:54:46
- [12] Materi Dosen. 2016. Pengertian dan Fungsi Kode ASCII, URL: <http://www.materidosen.com/2016/10/pengertian-dan-fungsi-kode-ascii-lengkap.html>, diakses tanggal 13 September 2016