

Aplikasi Kompresi File Dengan Metode Lempel-Ziv-Welch

Muhammad Iqbal Dzulhaq¹, Aan Ahmad Andayani²

¹Dosen STMIK Bina Sarana global, ²Mahasiswa STMIK Bina Sarana Global

Email : ¹miqbaldzulhaq@stmikglobal.ac.id

Abstrak— Dalam bidang teknologi informasi, komunikasi data sangat sering dilakukan. Komunikasi data ini berhubungan erat dengan pengiriman data menggunakan sistem transmisi elektronik dari satu terminal komputer ke terminal komputer yang lain. Besarnya ukuran data terkadang menjadi kendala dalam proses pengiriman data ini. Data dengan ukuran besar akan memakan waktu transfer yang lebih lama dibandingkan dengan data yang memiliki ukuran lebih kecil, terkadang ada resiko tidak dapat tertampung pada media penyimpanan dan tidak tersampainya, sehingga akan memperkecil kapasitas kosong dalam memori media penyimpanan. Oleh karena itu, manusia selalu berusaha untuk menemukan suatu cara alternatif untuk menangani permasalahan tersebut, salah satunya dengan cara kompresi. Penelitian dilakukan dengan menggunakan metode dictionary serta menggunakan algoritma Lempel-Ziv-Welch. Aplikasi kompresi ini ditekankan untuk file teks dan file gambar saja. Sedangkan keakuratan hasil kompresi dapat dilihat dari waktu kompresi dan rasio kompresi. Aplikasi kompresi akan dilakukan terhadap file yang mempunyai jenis dan ukuran yang berbeda-beda. Adapun hasil uji coba menunjukkan bahwa algoritma Lempel-ziv-welch lebih bagus digunakan dalam melakukan kompresi beberapa file teks karena ada file teks yang tidak bisa dilakukan kompresi. Akan tetapi hal ini juga sangat dipengaruhi oleh komposisi dari file yang akan dikompresi.

Kata kunci—file kompresi, algoritma Lempel-Ziv-Welch.

I. PENDAHULUAN

Salah satu kegunaan kompresi adalah untuk memperkecil kapasitas kosong dalam memori media penyimpanan, agar kita tidak terlalu boros menggunakan media penyimpanan tersebut. Kompresi data berarti suatu teknik untuk memampatkan data agar diperoleh data dengan ukuran yang lebih kecil daripada ukuran aslinya sehingga lebih efisien dalam menyimpannya serta mempersingkat waktu pertukaran data tersebut.

Pada kompresi data, terdapat dua tipe macam kompresi, yaitu lossless compression dan lossy compression. Pada lossless compression, semua informasi yang ada pada data akan kembali menjadi seperti aslinya dan tidak ada informasi yang hilang. Teknik ini biasanya digunakan untuk dokumen-dokumen, file executable, dan lainnya. Karena, kehilangan sebuah informasi merupakan hal yang fatal bagi file-file tersebut. Sedangkan pada lossy compression, tidak semua informasi yang ada akan kembali seperti semula. Hanya informasi-informasi inti yang dikembalikan. Hal ini terjadi, karena pada lossy compression informasi-informasi yang tidak berguna akan dihilangkan. walaupun ada informasi yang hilang, namun hal ini tidak terlalu disadari oleh pengguna.

Teknik ini biasanya digunakan pada file videos, gambar, suara yang mana file-file tersebut biasanya berukuran besar.

Dengan adanya kompresi diharapkan dapat menghemat biaya serta waktu yang dikeluarkan guna menambah fasilitas media penyimpanan data pada komputer serta mempercepat proses transfer data.

Beberapa software kompresi yang banyak digunakan para pengguna komputer saat ini diantaranya adalah WinZip (menghasilkan format.zip) dan WinRAR (menghasilkan format.rar) dan lainnya. Selain itu, telah banyak penelitian yang dilakukan mengenai kompresi file dengan berbagai macam algoritma. Maka berdasarkan latar belakang yang telah dikemukakan di atas, judul yang diambil dalam penelitian ini adalah “Aplikasi Kompresi File Dengan Metode Lempel-Ziv-Welch”.

II. LANDASAN TEORI

A. Penyimpanan Data

Perkembangan media penyimpanan data (data storage) sejak komputer tercipta berubah sangat signifikan. Perbandingannya sangat mencolok, sebagai contoh data yang tersimpan dalam sebuah media penyimpanan sangat kecil, di bawah 4096 bits. Penyimpanan data komputer, berasal dari bahasa Inggris "computer data storage" sering disebut sebagai memori komputer, merujuk kepada komponen komputer, perangkat komputer, dan media perekaman yang mempertahankan data digital yang digunakan untuk beberapa interval waktu. Penyimpanan data komputer menyediakan salah satu tiga fungsi inti dari komputer modern, yakni mempertahankan informasi. Ia merupakan salah satu komponen fundamental yang terdapat di dalam semua komputer modern, dan memiliki keterkaitan dengan mikroprosesor, dan menjadi model komputer yang digunakan semenjak 1940-an.

Penyimpanan data dilakukan untuk mencegah terjadinya kerusakan atau kehilangan data dan mempermudah membawa data-data tersebut ke manapun dan kapanpun dibutuhkan. Penyimpanan data pada komputer hanya dibatasi dengan kapasitas tertentu, sehingga dengan banyaknya data, memungkinkan tidak cukupnya perangkat keras penyimpan data untuk menyimpan semua data-data yang ada.

Dengan munculnya masalah itu, kita sebagai manusia yang mempunyai ilmu pengetahuan dan teknologi harus bisa berfikir dan memberikan solusi agar semua data yang ada dapat tertampung dalam sebuah penyimpan data. Salah satu ilmu pengetahuan yang dapat menyelesaikan masalah

itu adalah dengan membangun sebuah aplikasi kompresi data. Kompresi data ini berfungsi untuk memampatkan data yang berukuran besar, agar memperkecil kapasitas pada tempat penyimpanan data.

B. Data

Kata “data” diadopsi dari bahasa Inggris dan berasal dari kata Yunani “datum” yang berarti “fakta”. Data di komputer memiliki ukuran dalam penyebutannya. Data terkecil di komputer disebut dengan bit, yaitu sinyal elektronik yang melewati suatu rangkain digital (prosesor) komputer. Bit-bit tersebut selanjutnya dirangkai dan rangkaian tersebut diberi kode lagi yang disebut dengan character.

C. Kompresi Data

Kompresi data berarti sebuah proses mengkodekan informasi menggunakan bit atau information-bearing unit yang lain yang lebih rendah dari pada representasi data yang tidak terkodekan dengan suatu sistem encoding tertentu.

D. Jenis Teknik Kompresi

Teknik kompresi secara umum dapat diklasifikasikan menjadi tiga yaitu:

Pertama, Entropy coding adalah teknik kompresi yang menggunakan proses lossless. Tekniknya tidak berdasarkan pada media dengan spesifikasi dan karakteristik tertentu namun berdasarkan urutan data serta tidak memperhatikan semantik data.

Kedua, Source coding adalah teknik kompresi dengan menggunakan proses lossy. Teknik ini berkaitan dengan data semantik (arti data) dan media.

Ketiga, Hybrid coding adalah teknik kompresi dengan menggunakan kombinasi atau gabungan dari entropy coding dan source coding.

E. Rasio Kompresi

Proses kompresi adalah proses encoding yang menghasilkan data yang sudah dikompresi yang disebut aliran data encoded. Sebaliknya aliran data yang telah dikompresi harus dilakukan proses dekompresi untuk menghasilkan kembali aliran data yang asli. Karena proses dekompresi menghasilkan decoding dari aliran data yang sudah dikompresi maka hasilnya adalah aliran data decoded.

Tingkat pengurangan data yang dicapai sebagai hasil dari proses kompresi disebut rasio kompresi. Rasio ini merupakan perbandingan antara panjang data string asli dengan panjang data string yang sudah dikompresi, seperti dituliskan dalam persamaan berikut:

$$\text{Rasio} = (\text{Ukuran File Asli})/(\text{Ukuran File Terkompresi})$$

Jika dinyatakan dalam prosentase maka dituliskan dalam persamaan berikut:

$$P = (1 - (\text{Ukuran File Terkompresi})/(\text{Ukuran File Asli})) * 100\%$$

Yang berarti ukuran file berkurang sebesar P (dalam persentase) dari ukuran semula. Semakin tinggi rasio tingkat

suatu teknik kompresi data maka semakin efektif teknik kompresi tersebut. Pada saat dikompresi, rasio kompresi akan berselang-seling berdasarkan pengaruh data terhadap algoritma yang digunakan.

F. Metode Kompresi

Metode kompresi secara umum dapat diklasifikasikan menjadi tiga, yaitu:

1. Algoritma Huffman

Algoritma Huffman dikembangkan oleh David Huffman, seorang mahasiswa MIT. Jika ditinjau dari segi teknik pengkodean karakter yang digunakan, algoritma Huffman ini termasuk dalam algoritma yang menggunakan metode symbolwise. Yang dimaksud dengan metode symbolwise adalah suatu metode yang menghitung probabilitas kemunculan suatu karakter dalam suatu waktu.

2. Code Huffman

Code Huffman merupakan prefix code yang berisi himpunan bit biner dari suatu karakter yang tidak mungkin menjadi awalan (prefix) dari himpunan bit biner karakter lainnya. Untuk karakter yang sering muncul akan dikodekan dengan rangkaian bit yang pendek, sedangkan untuk karakter yang jarang muncul akan dikodekan dengan bit yang lebih panjang. Kode prefix biasanya direpresentasikan dalam sebuah pohon biner, di mana setiap cabangnya memiliki suatu label nilai tertentu.

3. Algoritma LZW

Lempel-ziv-Welch (LZW) adalah algoritma kompresi lossless universal yang diciptakan Abraham Lempel, Jacob Ziv, dan Terry Welch. Algoritma ini melakukan kompresi dengan menggunakan dictionary, di mana fragme-fragmen teks digantikan dengan indeks yang diperoleh dari sebuah kamus. Pendekatan ini bersifat adaptif dan efektif karena banyak karakter dapat dikodekan dengan mengacu pada string yang telah muncul sebelumnya dalam teks. Prinsip kompresi tercapai jika referensi dalam bentuk pointer dapat disimpan dalam jumlah bit yang lebih dibandingkan string aslinya.

Sebagai contoh, string “ABBABABAC” akan dikompresi dengan LZW. Isi dictionary pada awal proses diset dengan tiga karakter dasar yang ada: “A”, “B”, “C”. Tahapan proses kompresi ditunjukkan pada tabel dibawah ini :

Tabel 1 Tahapan proses kompresi

Langkah	Posisi	Karakter	Dictionary	Output
1	1	A	[4] A B	A
2	2	B	[5] B B	B
3	3	B	[6] B A	B
4	4	A	[7] A B A	AB
5	6	C	[8] A B AC	ABA
6	9	C	---	C

Prinsip umum kerja algoritma LZW adalah mengecek setiap karakter yang muncul kemudian menggabungkan dengan karakter selanjutnya menjadi sebuah string jika string baru tersebut tidak berada dalam dictionary atau belum diindekkan maka string baru tersebut akan diindekkan ke dalam dictionary.

G. UML (Unified Modelling Language)

UML (Unified Modeling Language) adalah sebuah bahasa untuk menentukan, visualisasi, konstruksi, dan mendokumentasikan artifact (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak. Artifact dapat berupa model, deskripsi atau perangkat lunak) dari system perangkat lunak, seperti pada pemodelan bisnis dan system non perangkat lunak lainnya.

III. ANALISIS DAN PERANCANGAN SISTEM

A. Analisis Sistem

Analisis ini diperlukan sebagai dasar bagi tahapan perancangan sistem. Analisis sistem Desain dan implementasi ini meliputi desain data, deskripsi sistem, desain proses dan implementasi desain dan semua yang diperlukan dalam aplikasi kompresi file.

B. Analisis Data

Dalam sistem pengkompresian file ini sekumpulan file-file yang mempunyai berbagai jenis ekstensi yang digunakan sebagai objek penelitian. Umumnya pengenalan tipe file tertera pada nama file tersebut, yaitu tiga huruf paling kanan setelah titik. Fungsinya adalah untuk mengetahui atau membedakan jenis file.

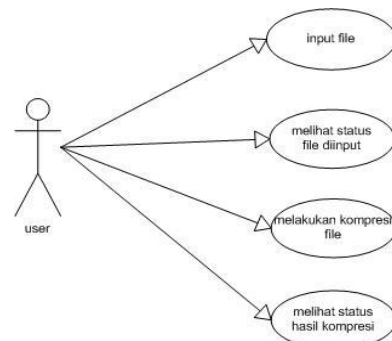
Metode yang digunakan pada sistem ini adalah dengan metode dictionary. Sedangkan tahapan dalam pengkompresian file ada 2 tahapan. Tahap pertama yaitu proses pengkompresian terhadap beberapa file dan tahap kedua yaitu proses dekomposisi terhadap file-file yang telah terkompresi.

Pada tahap kompresi, proses-proses yang dilakukan adalah :

- a. Dictionary (kamus) diinisialisasi dengan semua karakter dasar yang ada : {'A'..'Z', 'a'..'z', '0'..'9'}.
- b. W <- karakter pertama dalam stream karakter.
- c. K <- karakter berikutnya dalam stream karakter.
- d. Lakukan pengecekan apakah (W+K) terdapat dalam Dictionary.
- e. Jika ya, maka $W = W + K$ (gabungkan W dan K menjadi string baru).
- f. Jika tidak, maka :
 - Output sebuah kode untuk menggantikan string W.
 - Tambahkan string (W+ K) ke dalam dictionary dan berikan nomor/kode berikutnya yang belum digunakan dalam dictionary untuk string tersebut.
 - $W = K$.
- g. Lakukan pengecekan apakah masih ada karakter berikutnya dalam stream karakter.
 - Jika ya, maka kembali ke langkah b.
 - Jika tidak, maka output kode yang menggantikan string W, lalu terminasi proses (stop).
- h. Tulis output kode, hasil output dari proses kompresi adalah file dengan ekstensi *.LZW, dan akan disimpan dengan nama dan direktori sesuai dengan yang telah ditentukan oleh user.

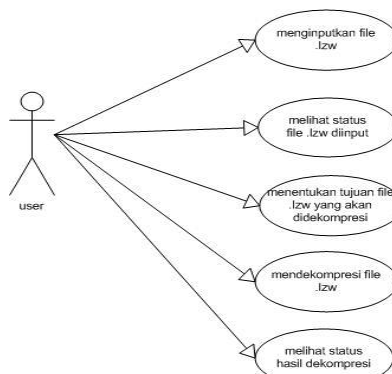
C. Use Case

Spesifikasi Use Case Kompresi



Gambar 1 Spesifikasi Use Case Kompresi

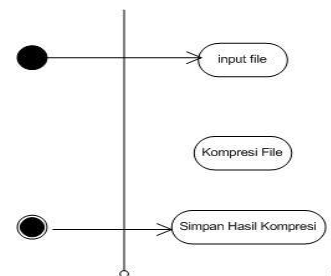
Spesifikasi Use Case Dekompresi



Gambar 2 Spesifikasi Use Case Dekompresi

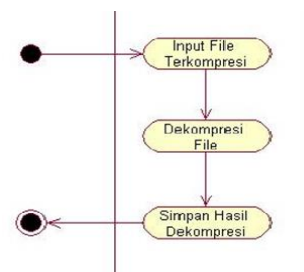
D. Activity Diagram

Activity Diagram Kompresi



Gambar 3.3 Diagram Sistem Kompresi

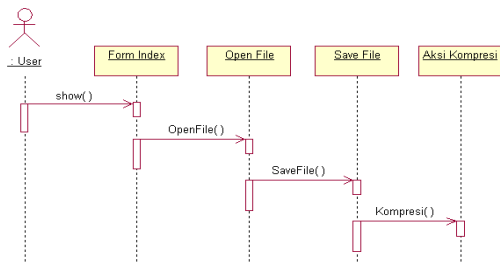
Activity Diagram Dekompresi



Gambar 3 Diagram Activity Sistem Dekompresi

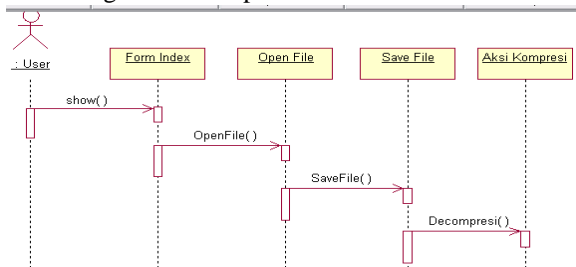
E. Sequence Diagram

Sequence Diagram Kompresi



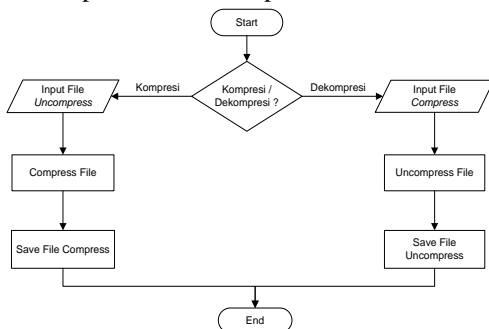
Gambar 4 Sequence Diagram Proses Kompresi

Sequence Diagram Dekompresi

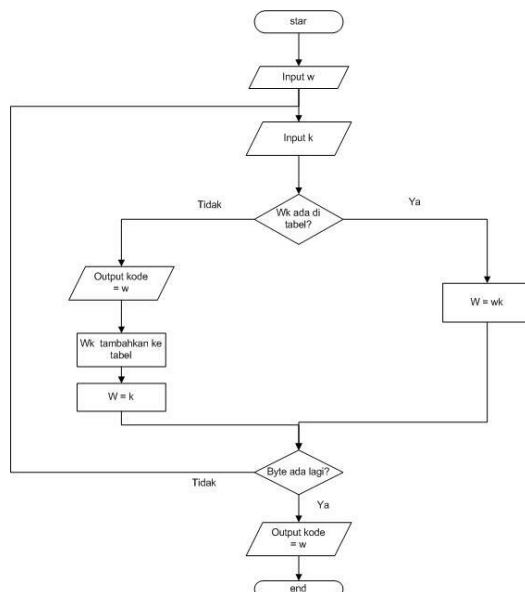


Gambar 5 Sequence Diagram Proses Dekompresi

Flowchart Kompresi dan Dekompresi File



Gambar 6 Flowchart Kompresi File



Gambar 7 Flowchart Algorithm Lempel-Ziv-Welch

F. Algoritma Lempel-Ziv-Welch untuk Kompresi

Pada algoritma ini, setiap byte hasil proses kompresi Flowchart Algoritma Lempel-Ziv-Welch dianggap sebagai 12 bit. Sehingga dictionary yang dipakai adalah sebesar 4096 (2¹²) dan dikurangi dengan 256 karakter ASCII. Karena indeks dimulai dari nol, kisaran indeks dictionary-nya dimulai dari 256 sampai dengan 4095.

Algoritma Flowchart Algoritma Lempel-Ziv-Welch melakukan pembacaan satu karakter k pada satu waktu. Karakter k dipasangkan dengan karakter sebelumnya yaitu w, setiap pasangan karakter ini dimasukkan ke dalam tabel. Lalu dibaca lagi karakter k, jika pasangan w dan k ada di dalam tabel maka w diisi oleh wk dan output kode adalah w, demikian seterusnya hingga aliran data habis.

G. Algoritma Lempel-Ziv-Welch untuk Dekompresi

Untuk melakukan dekomposisi, tidak diperlukan tabel terdahulu pada proses kompresi data, dan tabel tidak perlu diikuti dalam data kompresi, dekomposisi membuat sendiri tabelnya. Jika ada masukan data karakter k, maka output kode adalah k, selanjutnya nilai w diisi dengan karakter k. Ada satu variable tambahan yaitu entry yang digunakan untuk mengambil satu karakter dari k untuk dijadikan dictionary. Jika entry berisi 2 byte maka diambil byte pertama saja untuk kemudian ditambahkan dengan w kemudian dimasukkan dalam dictionary.

IV. KESIMPULAN

Kesimpulan yang diperoleh dari hasil penelitian tentang aplikasi kompresi file adalah sebagai berikut:

1. Penerapan metode dictionary dengan menggunakan algoritma Lempel-Ziv-Welch dalam aplikasi kompresi file ini memiliki waktu yang tidak terlalu cepat tergantung dari besar file yang tersedia.
2. Berdasarkan uji coba yang telah dilakukan, dengan melakukan pengkompresian terhadap beberapa file teks dan gambar, algoritma Lempel-Ziv-Welch lebih bagus digunakan dalam pengkompresian file teks namun ada beberapa file yang tidak bisa dilakukan kompresi seperti docx dan xlsx, apabila dilakukan kompresi file tersebut bertambah besar, setelah dilakukan dekomposisi ukuran menjadi normal kembali.

DAFTAR PUSTAKA

- [1] S. Dharwiyanti dan R. S. Wahono, "Pengantar Unified Modeling Language," 2003.
- [2] A. H. Basori, "Tutorial Rational Rose," 2003.
- [3] J. Kusnendar, "Diktat Mata Kuliah Rekayasa Perangkat Lunak," 2011.
- [4] Muhsin, "Diktat Mata Kuliah Riset Teknologi Informasi," 2010.
- [5] Linawati dan H. P. Panggabean, "Perbandingan Kinerja Algoritma Kompresi Huffman, Lzw, dan Dmc pada Berbagai Tipe File," 2004.
- [6] I. Haryanto dan Roy, "Kompresi Data dengan Algoritma Huffman dan Perbandingannya dengan Algoritma Lzw dan Dmc," 2009.
- [7] T. Suryana, "Metode RUP," STMIK LIKMI, Bandung, 2007.
- [8] M. M. Abdullah, "Kompresi String Menggunakan Algoritma Lzw dan Huffman," 2009.