

IMPLEMENTASI ALGORITMA BYTE PAIR ENCODING UNTUK KOMPRESI FILE

¹Ahmad Meidy Ruslida, ²Sapri, ³Devi Sartika

¹Mahasiswa, Universitas Dehasen Bengkulu, Bengkulu, Indonesia
Alamat (Kampus I: Jl. Meranti Raya No.32 Sawah Lebar Kota Bengkulu; e-mail:
ahmad.meidy.ruslida@gmail.com)

²Dosen Tetap Program Studi Informatika Fakultas Ilmu Komputer Universitas Dehasen Bengkulu
Kampus I: Jl Meranti Raya No.32 Sawah Lebar Kota Bengkulu 38228 Telp. (0736) 22027, Fax. (0736) 341139;
e-mail: sapri@unived.ac.id, devisartika@unived.ac.id)

(*received*: Juni 2022, *revised* : Agustus 2022, *accepted* : Oktober 2022)

ABSTRACT Data compression is a field that focuses on forming a small output file from a large file. The need for compression has given birth to several methods that can be implemented in communication activities on the network as well as on storage activities. The byte pair encoding algorithm is a compression algorithm that takes advantage of the emergence of repeated or frequent character byte pairs. The repeating character byte pairs are in the data and are substituted with a wildcard byte where the wildcard byte is not in the character byte series of the file. One of the well-known compression methods is the byte pair encoding compression method. The implementation of the system uses the Visual Basic 2010 programming language and the method used in this research is an applied research method.

Keywords: Compression, Byte Pair Encoding, Document.

Intisari Kompresi data atau data compression merupakan bidang yang berfokus pada membentuk output file dengan ukuran yang kecil dari file yang berukuran besar. Kebutuhan akan kompresi telah melahirkan beberapa metode yang dapat di-implementasikan pada kegiatan komunikasi pada jaringan maupun pada kegiatan penyimpanan. Algoritma byte pair encoding merupakan algoritma kompresi yang memanfaatkan munculnya pasangan byte karakter yang berulang atau frekuen. Pasangan byte karakter yang berulang tersebut di data dan disubstitusikan dengan sebuah byte karakter pengganti yang mana byte karakter pengganti tersebut tidak terdapat pada deret byte karakter dari file. Salah satu metode kompresi yang cukup dikenal adalah metode kompresi byte pair encoding. Implementasi sistem menggunakan bahasa pemrograman Visual Basic 2010 dan metode yang digunakan dalam penelitian ini adalah metode terapan (*applied research*)

Kata kunci : Kompresi, byte pair encoding, dokumen

I.PENDAHULUAN

Kemajuan ilmu pengetahuan dan teknologi khususnya komputer telah berinovasi dan menjadi kebutuhan primer bagi manusia. Inovasi – inovasi dalam bidang komputer banyak mengalami hambatan

dikarenakan alasan kapasitas penyimpanan atau memori yang masih sangat mahal. Inovasi perangkat lunak atau aplikasi menghadirkan dilema baru, yaitu semakin besar *size file* atau ukuran berkas yang dihasilkan dari berbagai sumber.

Ukuran berkas atau *file* yang besar bukan hanya menjadi masalah dalam penyimpanan, namun juga berkembang menjadi masalah dalam pertukaran data. Kompresi merupakan salah satu teknik pemadatan yang memanfaatkan substitusi terhadap byte atau karakter yang sering muncul dengan mensubstitusikan byte atau karakter tersebut dengan byte lain yang lebih sederhana sehingga ukuran hasil pemadatan jauh lebih kecil dibandingkan dengan ukuran asli. Dengan kata lain bidang kompresi data bertujuan untuk mengatasi masalah-masalah yang diakibatkan oleh besarnya ukuran atau kapasitas file. Dalam bidang kompresi data telah banyak muncul metode – metode yang dapat digunakan dalam kompresi data.

Salah satu metode kompresi data adalah metode *Byte Pair Encoding*, dimana metode ini merupakan metode tipe *lossless* yang artinya *file* atau berkas yang telah dikompresi dapat direkonstruksi ulang menjadi *file*

sebelum dikompresi. *Byte Pair Encoding* merupakan metode kompresi sederhana yang mana sederetan data yang memiliki karakter atau symbol yang sama diwakili oleh sebuah karakter yang tidak terdapat pada data tersebut.

Byte Pair Encoding adalah teknik kompresi yang memanfaatkan table atau pustaka acuan dalam menggantikan pasangan byte atau karakter sehingga pesan atau teks menjadi lebih singkat sehingga dapat disimpan dengan ukuran yang lebih kecil, pada tahap awal sebuah table acuan harus dibangun terlebih dahulu dengan mendata karakter – karakter yang tidak terdapat pada data yang akan dikompresi. *Byte Pair Encoding* merupakan salah satu metode

kompresi yang sederhana namun efektif dan cepat sehingga masih digunakan sampai sekarang.

1. Khususnya pada *file*.

II. TINJAUAN PUSTAKA

A. Pengertian Aplikasi

Aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output* (Budiman & Mulyani, 2016). Aplikasi merupakan penerapan, menyimpan sesuatu hal, data, permasalahan, pekerjaan ke dalam suatu sarana atau media yang dapat digunakan untuk diterapkan menjadi sebuah bentuk yang baru (Melani, Siregar, & Siregar, 2018)

B. Pengertian File

File merupakan data yang ada pada komputer. Setiap data yang ada pada komputer dapat dikategorikan sebagai *file*. *File* tidak hanya terbatas pada data-data tertentu saja. Setiap data baik itu data gambar, data angka, data kata, data video, data suara, data aplikasi, dan data-data lainnya merupakan sebuah *file*. *File* atau berkas adalah sebuah mekanisme untuk menyimpan program maupun data hasil olahan komputer agar dapat digunakan pada masa yang akan datang. (Suyanto, 2018).

File di komputer pada umumnya disimpan di dalam suatu folder tertentu tergantung dari pemilik komputer tersebut yang ingin dimana tempat menyimpannya, setiap *file* memiliki ekstensi masing-masing tergantung jenis *file* itu sendiri. Ekstensi *file* adalah sebagai tanda yang membedakan jenis-jenis dari file. Adapun beberapa ekstensi file itu adalah sebagai berikut :

Tabel 1 Jenis File

Nama File	Ekstensi File	Keterangan
File System	Sys, Com, Bak, Bat, Tmp, Dan Exe	File sistem ini berfungsi untuk menjalankan program di dalam komputer sesuai dengan peruntukannya, dan juga menjalankan berbagai aplikasi yang diinstal ke dalam komputer
File Video	Mpg, Wmv, Mp4, 3gp, avi, flv, KV	Masing-masing ekstensi ini menunjukkan bahwa masing-masing video memiliki jenis pemutar yang berbeda seperti Window Media Player
File Dokumen	Doc, Odt, Xls, Ods, Pdf, Ppt, Txt	Masing-masing ekstensi tersebut menunjukkan jenis file dokumennya, dan hanya bisa dibuka jika

		di dalam komputer terinstal software atau aplikasi yang sesuai. Seperti Ms. Words, Ms. Excel, Adobe Acrobat Reader
File Gambar	Jpg, Jpeg, Png, Gif, Tif, Dan Lain-Lain	Ada umumnya gambar yang dihasilkan oleh kamera digital ataupun kamera manual akan berekstensi jpg atau jpeg. Aplikasi yang bisa digunakan mage editor, seperti PhotoShop, PhotoPaint, Paint, CorelDraw, AutoCad.
File Suara	Wav, Mp3, Midi, dan Rm	Sama halnya dengan file komputer lainnya, tidak semua file suara dapat dibukakan dengan satu aplikasi

C. Pengertian Kompresi

Kompresi data berarti sebuah proses mengkodekan informasi menggunakan bit atau *information-bearing* unit yang lain yang lebih rendah daripada representasi data yang tidak terkodekan dengan suatu sistem encoding tertentu. Kompresi data juga diartikan sebagai teknik untuk memampatkan data agar diperoleh data dengan ukuran yang lebih kecil daripada ukuran aslinya sehingga lebih efisien dalam menyimpannya atau mempersingkat waktu pertukaran data tersebut (Utari, 2016)

Kompresi adalah pengubahan data kedalam bentuk yang memerlukan *bit* yang lebih sedikit, biasanya dilakukan agar data dapat disimpan atau dikirimkan dengan lebih efisien. Kebalikan dan proses kompresi, yaitu dekompresi. Dekompresi sendiri merupakan proses untuk mengembalikan data baru yang telah dihasilkan oleh proses kompresi menjadi data awal. Data yang telah dilakukan kompresi dapat digunakan jaringan yang lebih rendah yang sesuai dengan kapasitas data yang telah dilakukan proses kompres (Firmansah & Setiawan, 2015).

Tujuan utama dari kompresi data adalah untuk menghemat ruang penyimpanan dan bandwidth jaringan yang digunakan untuk transmisi data tersebut. Salah satu aspek dari kompresi data adalah penghilangan redundansi. Pengenalan dari redundansi yang ada melibatkan tahap – tahap yang disebut sebagai modeling. Setelah proses penghilangan redundansi, informasi akan ditransformasi menjadi kode – kode berdasarkan aturan – aturan yang digunakan. Berdasarkan uraian tersebut, maka proses

kompresi data terbagi menjadi dua tahap, yaitu *modeling* dan *coding*.

Terdapat banyak metode kompresi data yang telah muncul seiring penelitian yang telah dilakukan. Metode atau algoritma tersebut dapat diklasifikasikan kedalam kompresi secara fisik dan secara logika. Kompresi secara fisik melakukan proses pada bit – bit data dan tidak memperdulikan makna dari data tersebut. Sedangkan secara logika melakukan kompresi dengan mengganti konten dari data dengan konten lain yang mirip namun dengan ukuran yang lebih kecil.

Kompresi data dapat dibagi menjadi dua jenis yaitu *lossles* dan *lossy*. Kompresi *lossles* merupakan kompresi dimana data yang telah dikompresi dapat kembali menjadi data semula, sedangkan kompresi *lossy* akan menghasilkan data kompresi yang tidak dapat dikembalikan kembali menjadi data semula. Metode kompresi data pada dasarnya menganut pada beberapa konsep pengulangan yang terjadi pada data yang mana terbagi menjadi beberapa kategori sebagai berikut :

1. *Information Entropy*, konsep yang menunjukkan keacakan dan ketidakteraturan pada sebuah informasi.
2. *Probability Coding*, konsep yang menggunakan pengkodean probabilitas kemunculan bagian dari informasi.

D. Kompresi Lossy

Kompresi Lossy adalah suatu metode untuk mengkompresi data dan men-dekompresinya, data yang diperoleh mungkin berbeda dari yang aslinya tetapi cukup dekat perbedaannya. Lossy kompresi ini paling sering digunakan untuk kompres data multimedia (Audio, gambar diam) (Purwoto & Ardiyanto , 2014). Kebanyakan kompresi lossy dapat disesuaikan menjadi beberapa tingkat kualitas yang berbeda, yang mana nilai akurasi yang diperoleh berbanding terbalik dengan nilai kompresi hasil dari proses kompresi. Semakin baik nilai akurasi berarti data hasil kompresi dan data semula tidak mengalami perubahan yang signifikan, namun mengakibatkan tingkat kompresi yang tidak baik. Nilai akurasi yang rendah menyebabkan data mengalami perubahan yang signifikan namun menghasilkan tingkat kompresi yang cukup baik

F. Kompresi Lossless

Kompresi Lossless diperlukan untuk data teks dan file, seperti catatan bank, artikel teks dll. Format kompresi lossy mengalami generation loss yaitu jika melakukan berulang kali kompresi dan dekompresi file akan menyebabkan kehilangan kualitas secara progresif. hal ini berbeda dengan kompresi data lossless. ketika pengguna yang menerima file terkompresi secara lossy (misalnya untuk mengurangi waktu download) file yang diambil dapat sedikit berbeda dari yang asli dilevel bit ketika tidak

dapat dibedakan oleh mata dan telinga manusia untuk tujuan paling praktis (Purwoto & Ardiyanto , 2014).

Kompresi lossless merupakan kebalikan dari tipe kompresi lossy dimana kompresi lossless memungkinkan untuk rekonstruksi kembali ke data awal dimana kompresi lossy tidak. Namun kompresi lossless memiliki tingkat kompresi yang lebih rendah dari pada tipe kompresi lossy walaupun dari beberapa bentuk data yang berbeda bisa saja menghasilkan kebalikannya. Tipe kompresi lossless biasanya banyak digunakan pada data atau berkas yang bersifat data diskrit yang mana tiap entitas dari data merupakan informasi yang penting sehingga perlunya rekonstruksi ulang dari data yang telah di kompresi. Oleh karena itu tipe kompresi lossless biasanya digunakan pada data file seperti dokumen dan spreadsheet.

G. Algoritma Byte Pair Encoding

Algoritma Byte Pair Encoding merupakan sebuah algoritma kompresi teks sederhana yang didasarkan oleh substitusi pola (Shibata & Kida, 2007). Operasi dasar dari kompresi adalah substitusi sebuah karakter yang mana tidak muncul pada pesan untuk sepasang atau dua karakter yang berpasangan yang sering muncul pada pesan. Operasi akan dilakukan berulang – ulang sampai semua karakter telah digunakan atau tidak ada lagi pasangan karakter yang muncul.

Teks hasil kompresi terdiri dari dua bagian, yaitu table substitusi dan pesan yang telah disubstitusi. Proses dekompresi membutuhkan waktu yang sangat sedikit dan proses yang cepat, lebih lanjut dekompresi dapat dilakukan secara parsial mengingat kompresi hanya bergantung pada proses substitusi. Kelebihan ini menjadi keuntungan tersendiri dibandingkan dengan metode yang menggunakan kamus lainnya. Berikut beberapa kriteria dari algoritma byte pair encoding.

- a. Merupakan tipe metode multipass.
- b. Hanya mampu mengeliminasi beberapa tipe redundansi.
- c. Menggunakan *offline dictionary*.
- d. Merupakan tipe kompresi yang dipengaruhi oleh kosa kata.

III. METODOLOGI PENELITIAN

A. Metode Penelitian

Dalam melakukan penelitian ini, peneliti menggunakan metode terapan (*applied research*). Penelitian terapan dilakukan berkenaan dengan kenyataan-kenyataan praktis, penerapan, dan pengembangan ilmu pengetahuan yang dihasilkan oleh penelitian dasar dalam kehidupan nyata. Penelitian terapan berfungsi untuk mencari solusi tentang masalah-masalah tertentu. Tujuan utama penelitian terapan adalah pemecahan masalah sehingga hasil penelitian dapat dimanfaatkan untuk

kepentingan manusia baik secara individu atau kelompok.

B. Analisa Kompresi File Dengan Algoritma Byte Pair Encoding

Algoritma *Byte Pair Encoding* merupakan sebuah algoritma kompresi teks sederhana yang didasarkan oleh substitusi pola. Operasi dasar pada byte pair encoding adalah substitusi pasangan karakter dengan sebuah karakter yang mewakili pasangan karakter tersebut. Analisis metode byte pair encoding pada penelitian ini akan dilakukan dengan melakukan operasi kompresi dan dekompresi menggunakan sample atau contoh deret karakter seperti yang terlihat sebagai berikut :

“ABABCABCD”

1. Kompresi

Proses kompresi merupakan proses transformasi deret karakter menjadi deret karakter yang baru yang berukuran lebih kecil atau jumlahnya lebih sedikit dari semula. Menggunakan contoh deret karakter yang telah disebutkan sebelumnya operasi kompresi pada byte pair encoding dapat dijabarkan sebagai berikut :

- a. Cari pasangan karakter yang paling sering muncul dari deret karakter. Pada contoh deret karakter yang digunakan, maka diperoleh pasangan karakter yang paling sering muncul adalah pasangan “AB”
- b. Substitusi pasangan karakter “AB” dengan karakter lain yang tidak terdapat pada deret karakter asli. Karakter yang terdapat pada deret karakter sample adalah “A”, “B”, “C” dan “D”, sehingga dapat dipilih karakter “E” sebagai substitusi dari pasangan “AB”. Setelah proses substitusi maka diperoleh deret karakter sebagai berikut :

“EECECD”

- c. Daftarkan substitusi pada table substitusi
E = AB
- d. Berikutnya cari kembali pasangan karakter yang paling sering muncul, sehingga diperoleh pasangan “EC”
- e. Substitusi pasangan “EC” dengan karakter lain yang dapat menggunakan karakter “F”. Karakter “F” dapat digunakan karena tidak terdapat pada deret karakter yang akan dikompresi dan juga tidak terdapat pada table substitusi. Setelah proses substitusi maka diperoleh deret karakter sebagai berikut :

“EFFD”

- f. Daftarkan substitusi ke dalam table substitusi

E = AB
F = EC

- g. Karena tidak ada lagi pasangan yang kemunculannya lebih dari satu maka proses

kompresi berhenti. Setelah proses kompresi selesai maka diperoleh hasil kompresi sebagai berikut :

“EFFD”

Berdasarkan operasi kompresi diatas, maka dapat disusun proses kompresi secara tabular

Tabel 2 Proses Kompresi

Tahap	Karakter	Tabel Substitusi
0	“ABABCABCD”	-
1	“EECECD”	E = AB
2	“EFFD”	E = AB F = EC

2. Dekompresi

Proses dekompresi merupakan proses transformasi balik deret karakter yang telah dikompresi menjadi deret karakter yang asli menggunakan panduan table substitusi hasil dari kompresi deret karakter tersebut. Menggunakan contoh deret karakter yang telah dikompresi sebelumnya operasi dekompresi pada byte pair encoding dapat dijabarkan sebagai berikut :

- a. Baca acuan substitusi karakter pada table substitusi di tahap yang paling terakhir. Berdasarkan table substitusi yang diperoleh dari proses kompresi sebelumnya, maka diperoleh acuan substitusi yang terakhir adalah :

F = EC

- b. Berdasarkan acuan substitusi, substitusi balik karakter dengan pasangan karakter yang telah ditentukan
“EFFD” » “EECECD”
- c. Eliminasi acuan substitusi yang telah diproses dari table substitusi
- d. Baca acuan substitusi berikutnya sehingga diperoleh acuan substitusi sebagai berikut :
E = AB
- e. Lakukan substitusi pada deret karakter sesuai dengan acuan substitusi yang diperoleh
“EECECD” » “ABABCABD”
- f. Eliminasi acuan substitusi dari table substitusi
- g. Karena acuan substitusi telah habis, maka proses dekompresi berhenti sehingga diperoleh deret karakter asli kembali

“ABABCABD”

Berdasarkan operasi dekompresi diatas, maka dapat disusun proses dekompresi secara tabular

Tabel 3 Proses Dekompresi

Tahap	Karakter	Tabel Substitusi
0	“EFFD”	E = AB F = EC
1	“EECECD”	E = AB
2	“ABABCABD”	-

1. Kesalahan pada struktur data atau akses database.
 2. Kesalahan performansi.
 3. Kesalahan inisialisasi dan tujuan akhir
- Pengujian dilakukan dengan memberi masukan pada form yang tersedia dengan beberapa data yang dikategorikan dalam kategori data yang sah (sesuai dengan peruntukannya), dan data yang tidak sah (data yang berfungsi untuk mengeksploitasi sistem), Setelah itu tanggapan yang diberikan oleh sistem akan dicatat.

IV. HASIL DAN PEMBAHASAN

A. Hasil Aplikasi

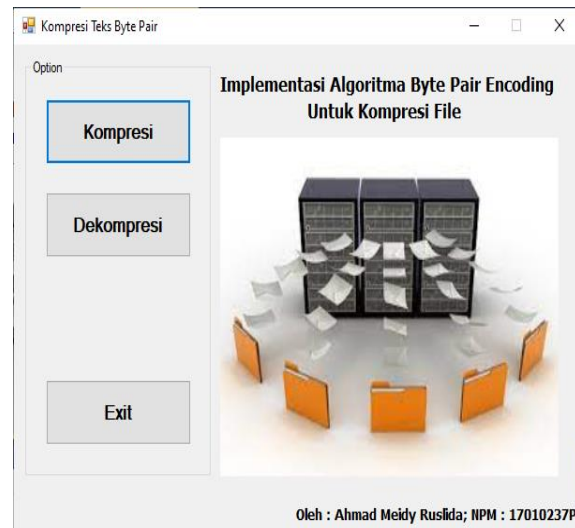
Aplikasi implementasi algoritma *byte pair encoding* untuk kompresi file dibangun sesuai dengan analisa dan perancangan seperti yang telah dijabarkan pada bab sebelumnya yaitu bab metodologi penelitian, maka pada bagian ini akan dipaparkan hasil dari aplikasi yang dibangun menggunakan perancangan yang telah di lakukan pada bab sebelumnya. Pada bab ini pembahasan akan dilakukan terhadap hasil dari sistem yang dibangun, fungsional sistem dan analisis terhadap kinerja sistem berdasarkan hasil *output* yang dihasilkan oleh sistem.

B. Implementasi Sistem

Pada aplikasi kompresi file dengan menggunakan algoritma *byte pair encoding* terdapat beberapa *interface* atau antarmuka yang di desain untuk mempermudah *user* atau pemakai dalam menggunakan atau menjalankan aplikasi ini. Adapun *interface* atau antarmuka adalah sebagai berikut :

1. Interface Menu Utama Aplikasi

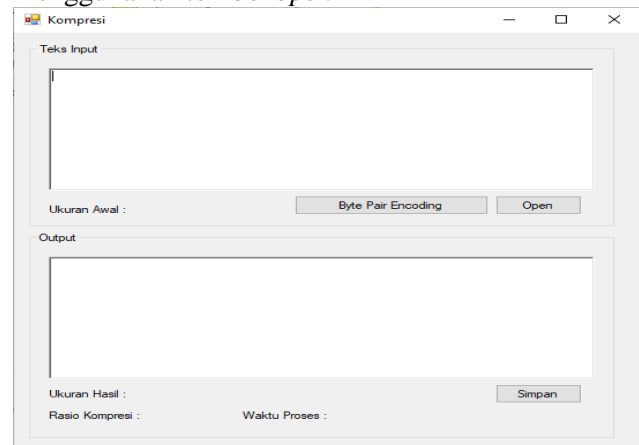
Interface menu utama merupakan *interface* yang pertama kali muncul ketika sistem ini dijalankan. Pada *interface* ini terdapat tombol pilihan yaitu tombol “*Kompresi*” yang digunakan untuk untuk masuk ke *interface* kompresi, pilihan tombol “*Dekompresi*” digunakan untuk masuk ke *interface* dekompresi dan pilihan tombol “*Exit*” yang digunakan untuk keluar dari sistem, dan terdapat label untuk judul sistem, *background* aplikasi serta label identitas pembuat sistem. Tampilan pada *form* menu utama dapat dilihat pada gambar 4.1



Gambar 1 Interface Menu Utama Aplikasi

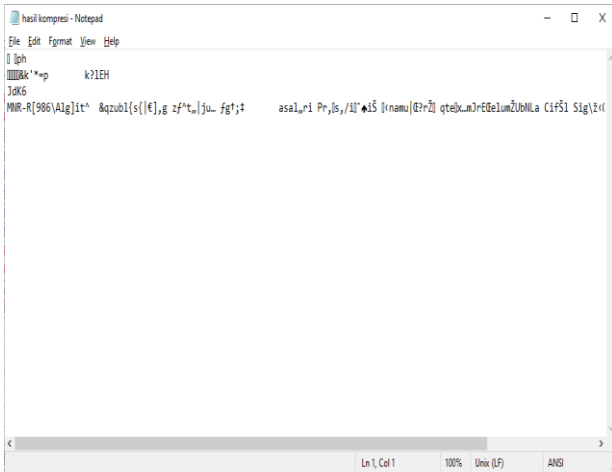
2. Interface Kompresi

Interface Kompresi akan ditampilkan jika *user* memilih tombol “*Kompresi*”. Secara garis besar *Interface* ini memiliki fungsi utama yaitu kompresi *file* yang merupakan *file* sumber dalam format *.txt* menjadi *file* hasil dengan ukuran yang lebih kecil. *Form* ini digunakan untuk melakukan kompresi terhadap *file* yang dipilih. Pengguna pertama sekali membuka *file* yang akan di-kompresi dengan menggunakan tombol *open*



Gambar 2 Form Kompresi

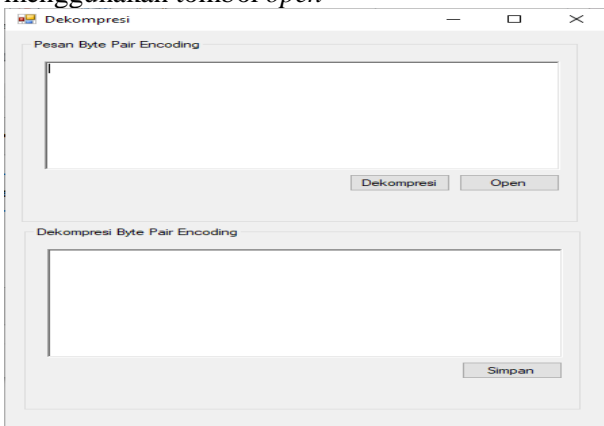
Proses kompresi kemudian dimulai dengan menggunakan tombol *byte pair encoding*. Hasil kompresi kemudian dapat disimpan kembali dalam format *file* menggunakan tombol *simpan*



Gambar 3 Hasil Kompresi

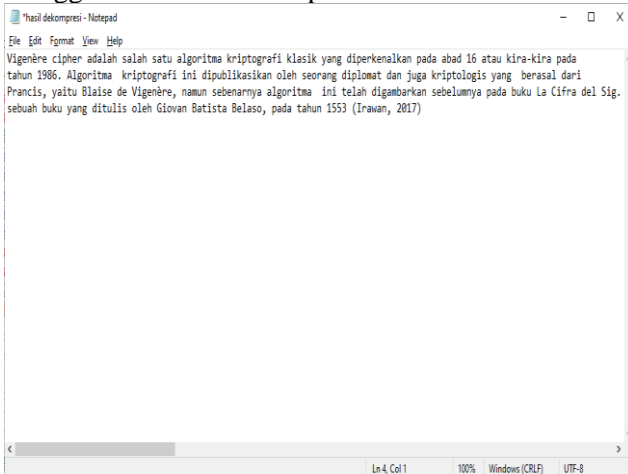
3. Form Dekompresi

Form ini digunakan untuk melakukan dekomposisi terhadap file yang dipilih. Pengguna pertama sekali membuka file yang akan di-dekompresi dengan menggunakan tombol *open*



Gambar 4 Form Dekompresi

Proses dekomposisi kemudian dimulai dengan menggunakan tombol dekomposisi. Hasil dekomposisi kemudian dapat disimpan kembali dalam format file menggunakan tombol simpan

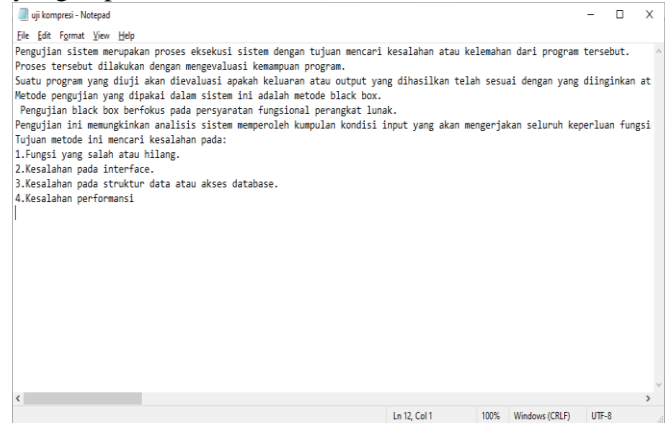


Gambar 5 Hasil Dekompresi

a. Pengujian Blackbox

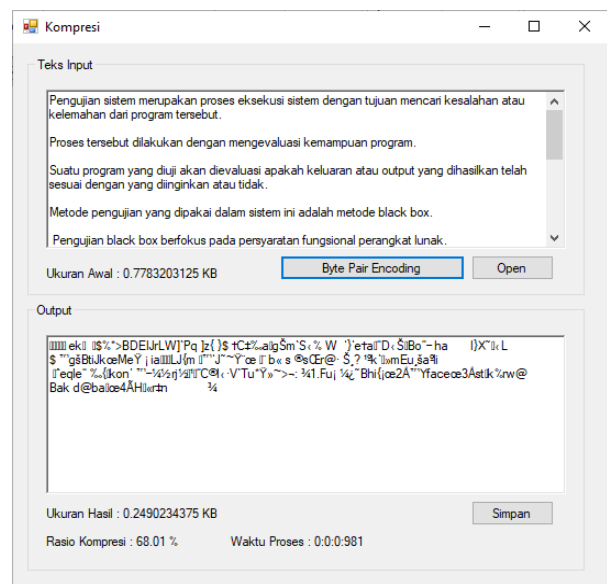
Pengujian *blackbox* dilakukan untuk memperoleh validasi terhadap fungsionalitas yang dimiliki oleh aplikasi yang dikembangkan. Pengujian *blackbox* pada penelitian ini dilakukan dengan menggunakan input teks yang akan dikompresi dan kemudian dikompresi kembali.

Input teks yang digunakan pada pengujian ini adalah input "uji kompresi.txt" yang mana memiliki teks yang dapat dilihat



Gambar 6 Input Pengujian Blackbox

Pengujian *blackbox* yang dilakukan pada penelitian ini dimulai dengan proses kompresi, kemudian dilanjutkan dengan melihat perbedaan ukuran berkas sebelum kompresi dan setelah kompresi, pengujian terus dilanjutkan dengan proses dekomposisi yang akan memeriksa apakah berkas atau teks dapat dikembalikan ke kondisi saat sebelum dikompresi. Pengujian kompresi pada pengujian ini yang dapat dilihat pada gambar 4.7 berikut.



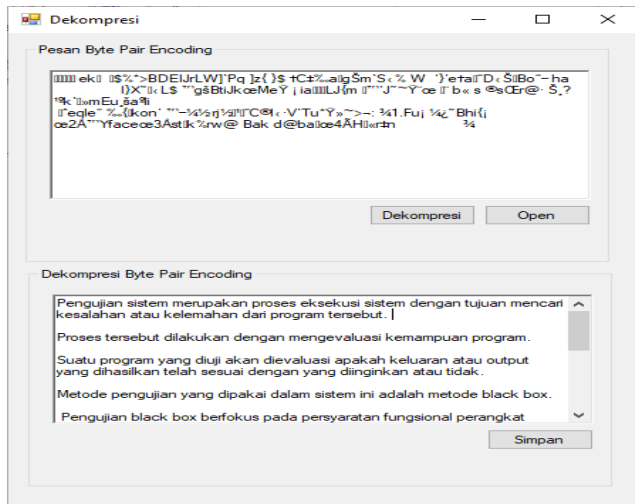
Gambar 7 Pengujian Kompresi

Hasil pengujian kompresi seperti yang terlihat pada gambar 4.7 dapat dilihat teks hasil kompresi telah disubstitusikan dengan karakter lain sehingga memiliki ukuran yang lebih kecil. Pada gambar

tersebut dapat dilihat bahwa ukuran file sebelum dikompresi 0.779 KB (pembulatan) dan 250 KB (pembulatan) setelah dikompresi.

Berdasarkan perbandingan kompresi seperti yang dapat dilihat pada gambar 4.7 dapat dilihat bahwa rasio kompresi dari aplikasi kompresi yang diperoleh adalah sebesar 66.01 %.

Pengujian *blackbox* dilanjutkan dengan melakukan dekomposisi kembali ke berkas teks asal yang dapat dilihat



Gambar 8 Pengujian Dekompresi

Pengujian dekomposisi seperti yang terlihat pada gambar 4.8 dapat dilihat proses dekomposisi balik terhadap berkas teks yang dikompresi sebelumnya dapat dikembalikan menjadi berkas teks semula dengan sempurna. Pengujian *blackbox* yang telah dilakukan mulai dari proses kompresi, perbandingan rasio kompresi dan proses dekomposisi menunjukkan bahwa aplikasi yang dikembangkan memiliki fungsi yang sesuai dan dapat beroperasi sesuai dengan yang diharapkan..

Dari hasil yang diperoleh, dapat dilihat bahwa program aplikasi kompresi menggunakan algoritma byte pair encoding yang dibangun dapat memberikan hasil yang cukup baik, dimana file yang di-kompresi dapat di-dekompresi kembali dengan sempurna.

V. PENUTUB

A. Kesimpulan

Berdasarkan hasil uji coba dan pembahasan program yang dilakukan, maka dapat di tarik Kesimpulanyaitu :

1. Telah dibangun aplikasi kompresi untuk meng-kompresi file menggunakan algoritma byte pair encoding. Pada proses pengujian yang dilakukan terhadap sistem, langkah awal dimulai dari memilih file yang akan di-kompresi, selanjutnya sistem akan melakukan kompresi pada file yang dipilih.
2. Algoritma byte pair encoding sangat baik diterapkan pada proses kompresi baik pada data

pesan maupun data file secara umum atau data binari, namun terdapat beberapa kelemahan pada algoritma byte pair encoding dimana jika isi file atau pesan memiliki persentase perulangan byte yang sedikit maka performa kompresi akan menurun drastis.

B. Saran

Saran-saran yang peneliti kemukakan diharapkan dapat lebih meningkatkan hasil yang telah didapatkan. Berikut beberapa saran yang disampaikan oleh peneliti :

1. Masih perlu adanya pengembangan dan penyempurnaan dari segi *graphic user interface* sehingga tampilan program tampak lebih menarik.
2. Mengenai saran yang digunakan khususnya komputer, perlu diadakan spesifikasi komputer yang sesuai dengan peneliti ajukan. Performa komputer yang buruk dapat mempengaruhi waktu yang dibutuhkan dalam proses kompresi dan dekomposisi

DAFTAR PUSTAKA

- [1] Aguilera, P. (2015). *Comparison of different image compression formats*. ECE 533 Project Report.
- [2] Alfina, O., & Harahap, F. (2019). Pemodelan Uml Sistem pendukung Keputusan Dalam Penentuan Kelas Siswa Siswa Tunagrahita. *METHOMIKA: Jurnal Manajemen Informatika & Komputerasi Akuntansi*, 143-150.
- [3] Budiman, A., & Mulyani, A. (2016). Rancang Bangun Aplikasi Sistem Informasi Persediaan Barang Di TB. Indah Jaya Berbasis . *Jurnal Algoritma Sekolah Tinggi Teknologi Garut ISSN: 2302-7339 Vol. 13 No. 2, 374-378*.
- [4] Firmansah, L., & Setiawan, E. B. (2015). Kompresi Data Audio Lossless Format Flac Menjadi Audio Lossy Format MP3 Dengan Algoritma Huffman Shift Coding. *e-Proceeding of Engineering*, 8067 - 8073.
- [5] Haviluddin. (2016). Memahami Penggunaan UML (Unified Modelling Language). *Jurnal Informatika Mulawarman*, 18-29.
- [6] Hidayatullah, P. (2014). *Visual Basic.Net Membuat Aplikasi Database Kreatif dan Program Kreatif*. Bandung: Informatika.
- [7] Melani, Siregar, Y. H., & Siregar, H. F. (2018). Perancangan Aplikasi Komik Hadist Berbasis Multimedia. *Perancangan Aplikasi Komik Hadist Berbasis Multimedia Vol.2, No.2 P-ISSN 2580-7927 / E-ISSN 2615-2738*, 113-121.
- [8] Nuraini, R. (2015). Desain algoritma Operasi Perkalian Matriks Menggunakan Metode Flowchart. *Jurnal Teknik Komputer*, 144 -151.

- [9] Purwoto, B., & Ardiyanto, D. (2014). Kompresi Citra Dengan Menggunakan Metode Delta Modulation. *Jurnal Emitter Vol. 14No. 01 ISSN 1411-8890*, 1-11.
- [10] Suendri. (2018). Implementasi Diagram UML (Unified Modelling Language) Pada Perancangan Sistem Informasi Remunerasi Dosen Dengan Database Oracle (Studi Kasus: UIN Sumatera Utara Medan). *ALGORITMA: Jurnal Ilmu Komputer dan Informatika*, 1-9.
- [11] Suyanto, Y. (2018). *Pemrograman Terstruktur Dengan Delphi*. Yogyakarta: Gadjah Mada University Press.
- [12] Utari, C. (2016). Implementasi Algoritma Run Length Encoding Untuk Perancangan Aplikasi Kompresi Dan Dekompresi File Citra. *Jurnal TIMES, Vol. V No 2 : 24-31 ISSN : 2337 - 3601*, 24-31.