Paper

Analisis Perbandingan Kompresi Half Byte ddan Byte Pair Encoding Pada File Bynari Atau Teks

Author: Dedek Prasetyo, Tommy, Rizko Liza

ISBN: 978-623-91911-2-2

Analisis Perbandingan Kompresi *Half Byte* dan *Byte Pair Encoding* Pada File Bynari Atau Teks

Dedek Prasetyo[1], Tommy[2], Rizko Liza[3]

[1,2,3] Universitas Harapan, Medan, Indonesia

[1]dedekprasetyo0111@gmail.com, [2]tomshirakawa@gmail.com, [3]risko.liza@gmail.com

Abstrak-File teks merupakan file yang berisi informasi-informasi dalam bentuk teks. Data yang berasal dari dokumen pengolah kata, angka yang digunakan dalam perhitungan, nama dan alamat dalam basis data merupakan contoh masukan data teks yang terdiri dari karakter, angka dan tanda baca. Proses berkirim pesan atau teks kini sudah menjadi kebutuhan setiap orang melalui berbagai platform media yang ada sebagai bentuk pesan visual. Besarnya ukuran transmisi teks tentu terkadang menjadi kendala, belum lagi bandwith juga menjadi masalah bagi pengguna dalam proses pengiriman. Salah satu solusi untuk menyelesaikan permasalahan tersebut ialah dengan memanfaatkan teknik pemampatan atau kompresi. Kompresi data adalah sebuah metode yang mendukung teknologi dalam pemanfaatan jaringan untuk pengiriman data, akses data, maupun penggunaan data. Beberapa dari sekian banyak algoritma kompresi yang dinyatakan baik untuk diterapkan pada kompresi data berbasis teks adalah Byte Pair dan Half Byte. Algoritma Half-Byte meman-faatkan empat bit sebelah kiri yang sering sama secara berurutan terutama pada file-file text. Algoritma Byte Pair Encoding merupakan sebuah algoritma kompresi teks sederhana yang didasarkan oleh substitusi pola. Kedua algoritma ini memandaatkan kemiripan data sebagai proses kompresi dengan menggunakan sebuah program yang telah dibuat dengan bahasa pemrograman visual basic 2010. Visual Basic 2010 adalah sebuah bahasa pemrograman berbasis OOP atau object oriented programming yang memanfaatkan teknologi .NET yang digunakan untuk membuat aplikasi di lingkungan kerja berbasis Windows. Visual Basic 2010 atau Visual Basic . Analisis perbandingan akan dilakukan berdasarkan hasil kompresi, rasio, dan waktu proses dalam penentuan algoritma mana yang lebih optimal.

Kata Kunci: *Kompresi, Teks, Byte Pair, Half Byte, Visual Basic 2010*

Abstract- A text file is a file that contains information in text form. Data derived from word processing documents, numbers used in calculations, names and addresses in the database are examples of text data input consisting of characters, numbers and punctuation marks. The process of sending messages or texts has now become a necessity for everyone through various media platforms that exist as a form of visual message. The large size of the text transmission of course sometimes becomes an obstacle, not to mention bandwidth is also a problem for users in the sending process. One solution to solve this problem is to use compression or compression techniques. Data compression is a method that supports technology in the use of networks for data transmission, data access, and data use. Some of the many compression algorithms that are declared good to be applied to text-based data compression are Byte Pair and Half Byte. The Half-Byte Algorithm takes advantage of the four left-hand bits which are often the same in sequence, especially in text files. *Byte Pair Encoding* Algorithm is a simple text compression algorithm based on pattern substitution. Both of these algorithms mandate the similarity of data as a compression process using a program that has been created with the Visual Basic 2010 programming language. Visual Basic 2010 is an OOP-based programming language or object oriented programming that utilizes .NET technology which is used to create applications in a work environment based on Windows. Visual Basic 2010 or Visual Basic . Comparative analysis will be carried out based on the results of compression, ratio, and processing time in determining which algorithm is more optimal.

Keywords: *Compression, Text, Byte Pair, Half Byte,Visual Basic*

1. PENDAHULUAN

Kompresi data adalah sebuah metode yang mendukung teknologi dalam pemanfaatan jaringan untuk pengiriman data, akses data, maupun penggunaan data. Pemanfaatan kompresi data dalam teknologi dan jaringan terjadi baik dengan kesadaran pemakai maupun tidak. Bahkan kompresi data secara langsung sudah menjadi bagian yang tidak terpisahkan dari setiap kebutuhan manusia akan jaringan, informasi, dan teknologi. Kompresi data sendiri

ISBN: 978-623-91911-2-2

merupakan sebuah metode untuk memadatkan (read mengecilkan) data dengan beberpa ketentuan serta algoritma yang unik dengan tujuan untuk mempermudah transportasi, akses, dan penyimpanan data didalam maupun diluar jaringan [1].Salah satu point utama dari sebuah kompresi data adalah efesiensi ukuran, kecepatan, serta kemampuan dari algoritma kompresi data yang digunakan terhadap data yang akan dikompresi. Sehingga seringnya terjadi perbandingan algoritma kompresi sebelum akhirnya pengguna menentukan algoritma kompresi data yang akan digunakan [2].

Pada penelitian sebelumnya dinyatakan, dari kedua jenis algoritma tersebut yang memiliki rasio (rasio) yang lebih besar sebagai tolak ukur dalam kinerja manipulasi data. Dengan kinerja yang baik maka akan dapat menghemat penggunaan ruang memori pada repositori dan cepat dalam mentransmisikan data dalam berkomunikasi antara user interface dengan repositori dalam manipulasi data [2].

Pada penelitian sebelumnya dinyatakan, dengan menggunakan teknik measure untuk hasil evaluasi akurasi maksimum 54.6% pada algoritma BPE dengan nilai recall maksimum 61.4% dan nilai precision maksimum 49.3%. Sedangkan algoritma Unigram (LM) menghasilkan akurasi maksimum 87.0% dengan nilai recall maksimum 90.1% dan nilai precision maksimum 84.1%. Sehingga dapat di simpulkan bahwa ukuran vocab size terhadap algoritma BPE dan Unigram dapat mempengaruhi hasil akurasi, masing-masing [3].

Dua diantara sekian banyak algoritma kompresi data yang sering digunakan adalah *Half Byte* dan *Byte Pair Encoding*. Kedua algoritma ini adalah beberapa dari sekian banyak algoritma kompresi data dengan jenis data teks yang sering digunakan dalam upaya pengguna meng-kompresi data sebelum dikirim atau sekedar penyimpanan saja. Tujuan penelitian ini yaitu Analisis Perbandingan Kompresi *Half Byte* dan *Byte Pair Encoding* Pada File Bynari Atau Teks dan untuk meneliti kemampuan masing-masing metode dalam kompresi data teks.

2. METODE PENELITIAN

2.1 File Teks

File teks merupakan file yang berisi informasi-informasi dalam bentuk teks. Data yang berasal dari dokumen pengolah kata, angka yang digunakan dalam perhitungan, nama dan alamat dalam basis data merupakan contoh masukan data teks yang terdiri dari karakter, angka dan tanda baca. Beberapa file teks menggunakan ekstensi file .TXT dan tidak mengandung gambar apa pun, tetapi yang lain mungkin berisi gambar dan teks tetapi masih disebut file teks atau bahkan disingkat sebagai 'file txt', jenis lain dari file teks adalah file 'teks biasa'. Ini adalah file yang berisi nol format (tidak seperti file RTF), artinya tidak ada yang tebal, miring, bergaris bawah, berwarna, menggunakan font khusus, dll. Beberapa contoh format file teks biasa termasuk yang berakhiran XML, REG, BAT, PLS, M3U, M3U8, SRT, IES, AIR, STP, XSPF, DIZ, SFM, TEMA, dan TORRENT [4].

2.2 Kompresi Data

Kompresi data merupakan proses untuk menghasilkan representasi digital yang padat atau mampat (compact) namun tetap dapat mewakili kualitas informasi yang terkandung pada data tersebut. Kompresi data dalam konteks komputer sains merupakan sebuah seni dalam mewakili informasi dalam bentuk mampat [5]. Kompresi Data Juga dijelaskan yang bertujuan untuk meminimalkan jumlah bit yang diperlukan untuk merepresentasikan suatu data [6].

Kompresi data bertujuan untuk mendesain sebuah algoritma kompresi untuk:

1. Mewakili data dalam ukuran yang lebih kecil.
2. Menghilangkan kelebihan (redundancy) data.
3. Mengimplementasi algoritma kompresi, untuk kompresi maupun dekompresi.

2.3 Algoritma *Half Byte* Encoding

Diketahui Peneliti sebelumnya Algoritma Half –Byte dapat memanfaatkan empat bit sebelah kiri yang sering sama secara berurutan terutama pada file teks [7]. Selain itu Peneliti sebelumnya juga menjelaskan bahwa NA Algoritma Half-Byte meman-faatkan empat bit sebelah kiri yang sering sama secara berurutan terutama pada file-file text. Dasar pemikiran algoritma ini adalah bahwa setiap karakter ASCII biasanya diwakili oleh 8 bit [8]. Misalnya pada suatu file text berisi tulisan “mengambil”, dalam heksadesimal dan biner karakter-karakter tersebut diterjemahkan.

Tabel 1. Contoh Konversi Teks Ke Dalam Biner

Karakter	Heksadesimal	Biner
m	6D	01101101
e	65	01100101

ISBN: 978-623-91911-2-2

n	6E	01101110
g	67	01100111
a	61	01100001
m	6D	01101101
b	62	01100010
i	69	01101001
l	6C	01101100

Pada tabel 1. menjelaskan bahwa pasangan byte dari sequence yang paling sering muncul adalah ”aa”, sehingga pasangan byte ”aa” digantikan oleh suatu byte yang tidak digunakan pada dalam data, misalnya dengan karakter ”Z”. Setelah dilakukan pergantian tabel ”aa” oleh ”Z”, maka data menjadi:”ZabZabac”. Kemudian dalam hal ini byte data ”ab” juga paling sering muncul, maka dilakukan pergantian dengan suatu byte data yang juga tidak dipakai, misalnya pada karakter ”Y”. Sehingga di peroleh hasil: ”ZYZYac”. Masih memungkinkan kemunculan paling sering pada pasangan byte ”ZY”, yang akan digantikan juga dengan suatu byte yang belum pernah digunakan sebelumnya, misalnya pada karakter ”X”, sehingga diperoleh hasil “XXAC”. Data ini tidak dapat dikompresi lebih lanjut dengan pengkodean pasangan byte karena tidak ada pasangan byte yang terjadi lebih dari sekali. Untuk mendekompresi data, cukup lakukan penggantian dalam urutan terbalik.

2.4 Algoritma *Byte Pair Encoding*

Algoritma *Byte Pair Encoding* merupakan sebuah algoritma kompresi teks sederhana yang didasarkan oleh substitusi pola [3]. 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. Misalkan data yang akan dikodekan adalah “aaabaaabac” Pasangan byte "aa" paling sering terjadi, sehingga akan diganti dengan byte yang tidak digunakan dalam data, yaitu "Z". kemudian dibuat tabel data dan pengganti seperti berikut:

Tabel 2. Data Kompresi *Byte Pair Encoding*

Iteration	Sequence	Penggantian
0	aaabaaabac	…
1	ZabZabac	{Z ←aa}
2	ZYZYac	{Y ←ab}
3	XXac	{X ←ZY}

Pada tabel 2. menjelaskan bahwa pasangan byte dari sequence yang paling sering muncul adalah ”aa”, sehingga pasangan byte ”aa” digantikan oleh suatu byte yang tidak digunakan pada dalam data, misalnya dengan karakter ”Z”. Setelah dilakukan pergantian tabel ”aa” oleh ”Z”, maka data menjadi:”ZabZabac”. Kemudian dalam hal ini byte data ”ab” juga paling sering muncul, maka dilakukan pergantian dengan suatu byte data yang juga tidak dipakai, misalnya pada karakter ”Y”. Sehingga di peroleh hasil: ”ZYZYac”. Masih memungkinkan kemunculan paling sering pada pasangan byte ”ZY”, yang akan digantikan juga dengan suatu byte yang belum pernah digunakan sebelumnya, misalnya pada karakter ”X”, sehingga diperoleh hasil “XXAC”.Data ini tidak dapat dikompresi lebih lanjut dengan pengkodean pasangan byte karena tidak ada pasangan byte yang terjadi lebih dari sekali. Untuk mendekompresi data, cukup lakukan penggantian dalam urutan terbalik.

3. HASIL DAN PEMBAHASAN

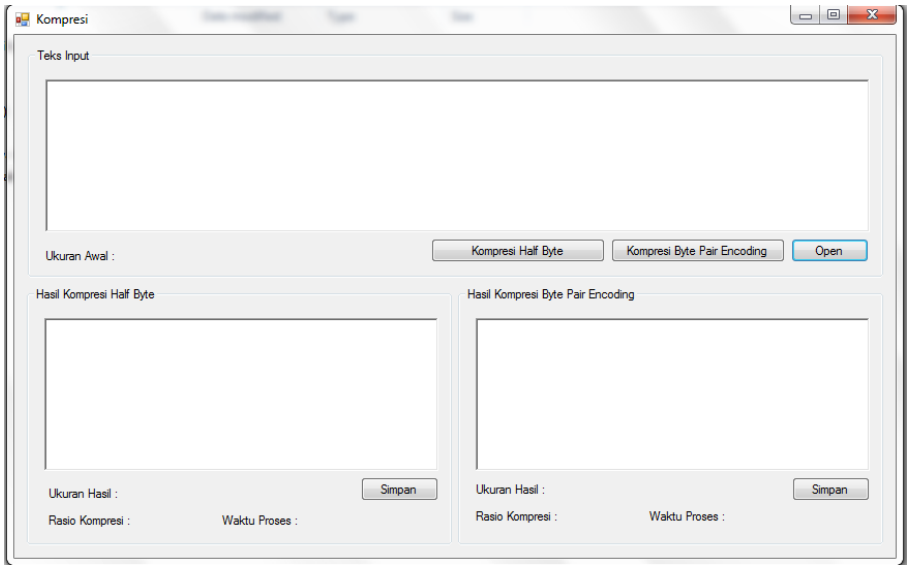
3.1 Perancangan Interface

Berikut adalah tampilan yang digunakan dalam aplikasi analisis perbandingan kompresi *Half Byte* dan *Byte Pair Encoding* pada *file* bynari atau teks. Program ini sendiri dibuat dengan meletakan semua elemen yang digunakan dalam proses kompresi data bynari atau teks dengan menggunakan algortima Half-Byte dan dan *Byte Pair Encoding*. Berikut tampilan dari aplikasi tersebut:

ISBN: 978-623-91911-2-2

Gambar 1. Halaman Awal Dan Pemilihan Kompresi

Pada gambar 1 pengguna telah memilih halaman kompresi selanjutnya pengguna akan dipindahkan kehalaman kompresi, pada halaman ini pengguna dapat memasukkan teks secara manual ataupun pengguna dapat memilih file txt yang sudah terlebih dahulu dipersiapkan untuk proses kompresi.



Gambar 2. Halaman Awal Dan Pemilihan Kompresi

Pada gambar 2 terlihat di halaman ini pengguna juga dapat memilih salah satu ataupun kedua algoritma kompresi sekaligus. Untuk proses perbandingan algortima yang akan diuji oleh penulis, maka penulis akan menggunakan kedua algoritma secara sekaligus. Pertama penulis akan mengambil file txt yang sudah dipersiapkan sebelumnya untuk proses kompresi.

3.2 Hasil Pengujian Algortima Half-Byte Dan *Byte Pair Encoding*

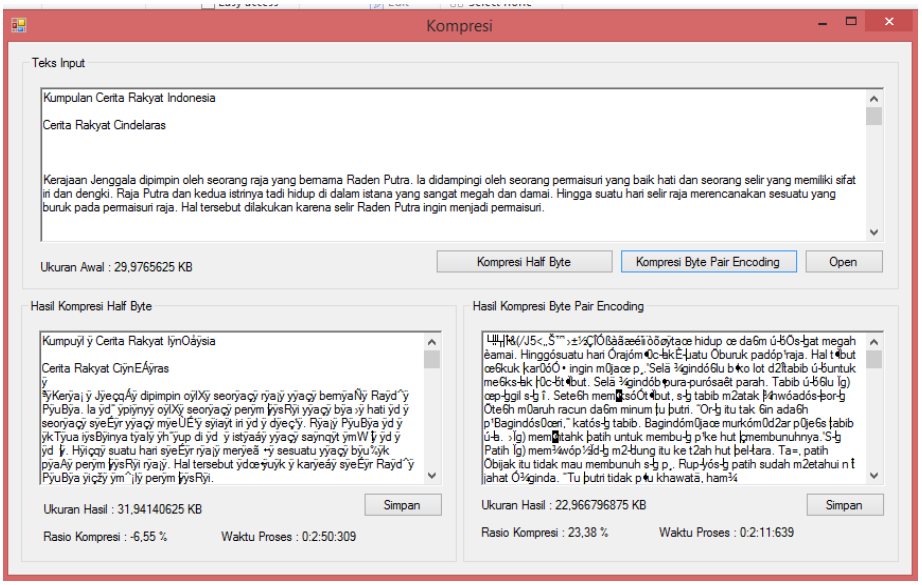
Pengujian dilakukan dengan melakukan kompresi dan dekompresi pada beberapa file teks dan kemudian akan dilanjutkan dengan pembahasan terhadap hasil pengujian yang dilakukan. Pada tabel 3.1 dapat dilihat file pengujian yang akan digunakan pada pengujian terdiri dari sepuluh buah file yang memiliki isi dan ukuran yangberbeda-beda. Proses pengujian dilakukan dengan melakukan kompresi dan dekompresi pada file-file berikut secara bergantian.

ISBN: 978-623-91911-2-2

Tabel 3. File Pengujian

No.	Nama File	Ukuran
1.	Pengujian 1.txt	29,9765 Kb
2.	Pengujian 2.txt	22,1328 Kb
3.	Pengujian 3.txt	19,4521 Kb
4.	Pengujian 4.txt	21,1972 Kb
5.	Pengujian 5.txt	20,8056 Kb
6.	Pengujian 6.txt (file biner)	61,4638 Kb
7.	Pengujian 7.txt (file biner)	21,2958 Kb
8.	Pengujian 8.txt (file biner)	23,0039 Kb
9.	Pengujian 9.txt (file biner)	19,0039 Kb
10.	Pengujian 10.txt (file biner)	19,0048 Kb

Pada gambar 3 dapat dilihat proses kompresi dari metode *Half Byte* dan *Byte Pair Encoding* dimana file awal yang berukuran 29,9765 Kb akan dikompresi dengan menggunakan kedu metode tersebut sehingga didapatkan hasil kompresi dengan menggunakan metode *Half Byte* menghasikan ukuran kompresi sebesar 31,9765 Kb dengan rasio -6,55% sedangkan pada metode *Byte Pair Encoding* menghasilkan ukuran kompresi sebesar 22,9967 Kb dengan rasio 23,38%.



Gambar 3. Halaman Kompresi *Half Byte* dan *Byte Pair Encoding*

Pengujian yang dilakukan pada penelitian ini bertujuan menganalisa metode kompresi yang digunakan dalam mengecilkan ukuran pesan teks menggunakan metode *Half Byte* dan *Byte Pair Encoding*. Berikut hasil pengujian kompresi data yang menggunakan data dari *notepad* dengan format .txt yang dapat kita lihat pada tabel 4.

ISBN: 978-623-91911-2-2

Tabel 4. Pengujian *File Teks*

No	Nama File	Ukuran	Half Byte		*Byte Pair Encoding*	
			Rasio Kompresi	Ukuran Kompresi	Rasio Kompresi	Ukuran Kompresi
1	Pengujian 1.txt	29,9765 Kb	31,9414 Kb	-6,55%	22,9967 Kb	23,38%
2	Pengujian 2.txt	22,1328 Kb	23,5634 Kb	-6,46%	16,5498 Kb	25,23%
3	Pengujian 3.txt	19,4521 Kb	20,4726 Kb	-5,25%	14,2441 Kb	26,77%
4	Pengujian 4.txt	21,1972 Kb	22,3857 Kb	-5,61%	15,5449 Kb	26,67%
5	Pengujian 5.txt	20,8056 Kb	22,2402 Kb	-6,9%	14,9521 Kb	28,13%
6	Pengujian 6.txt (file biner)	61,4638 Kb	49,4023 Kb	19,62%	12,7822 Kb	79,2%
7	Pengujian 7.txt(file biner)	21,2958 Kb	17,1621 Kb	19,41%	3,5517 Kb	83,32%
8	Pengujian 8.txt (file biner)	23,0039 Kb	18,5 Kb	19,58 %	4,3535 Kb	81,07%
9	Pengujian 9.txt (file biner)	19,0039 Kb	15,2226 Kb	19,9%	4,3330 Kb	77,2%
10	Pengujian 10.txt (file biner)	19,0048 Kb	15,2617 Kb	19,7%	4,1572 Kb	78,13%

4. KESIMPULAN

Dalam perancangan, pembuatan, dan pengujian aplikasi Analisis Perbandingan Kompresi *Half Byte* dan *Byte Pair Encoding* Pada File Bynari Atau Teks terdapat beberapa kesimpulan yang dapat ddiambil oleh penulis, diantaranya adalah sebagai berikut :

1. Algoritma Kompresi *Half Byte* dan *Byte Pair Encoding* telah terbukti mampu meng-kompresi data text dengan cukup baik dan dapat diimplementasikan kedalam sistem tanpa adanya kesalahan dalam penggunaan sistem baik pada saat proses kompresi maupun dekompresi data text.
2. Analisa hasil kompresi *Half Byte* dan *Byte Pair Encoding* berdasarkan waktu, hasil kompresi, dan juga size atau ukuran data hasil kompresi menunjukkan bahwasanya algoritma *Byte Pair Encoding* memiliki efisiensi yang lebih tinggi dibandingkan Half-Byte encoding dalam hal waktu kompresi, hasil kompresi, dan juga size atau ukuran data hasil kompresi data text yang berisi karakter teks dan juga data yang berisi bilangan biner.
3. Tergantung dari karakter yang digunakan pada sebuah data teks, ternyata dapat mempengaruhi sebuah file hasil kompresi dari masing-masing algoritma. Semakin banyak pasangan yang ditemukan pada file akan lebih meningkatkan efesiensi hasil kompresi *Byte Pair Encoding*, dan Half-Byte encoding yang mengutamakan kesamaan byte karakter dapat mengalami membesarnya size data dikarenakan tidak adanya byte yang sama pada karakter dan kemudian ditambah dengan bit penanda sehingga data hasil kompresi lebih besar dibandingkan file asli (kompresi gagal).
4. Setelah dilakukan pengujian terhadap kedua metode diperoleh rasio terbaik dari metode Byte Pair dari pengujian file berisi teks maupun bilangan biner. Bukan berarti Hakf Byte tidak baik dalam hal kompresi, Algorita *Half Byte* juga dapat melalukan komresi yang baik pada file berisi biner sehingga dapat disimpulkan bahwa metode Byte Pair lebih baik dibandingkan metode Half Byte.

ISBN: 978-623-91911-2-2

DAFTAR PUSTAKA

[1] E. S. Panggabean, "Analisa Perbandingan Algoritma Lempel Ziv Welch Dan Algoritma Deflate Pada File Teks Dengan Metode Independent Sample T-Test," *J. Pelita Inform.*, vol. 17, pp. 79–82, 2018.

[2] Supriyadi and O. Frida, "Analisis Perbandingan Pemampatan Data Teks Dengan Menggunakan Metode Huffman Dan Half – Byte," *Algoritm. J. Ilmu Komput. dan Inform.*, vol. 2, no. 1, pp. 1–6, 2018.

[3] T. Jamaluddin, M. A. Bijaksana, and I. Asror, "Perbandingan Algoritma Sentencepiece BPE dan Unigram Pada Tokenisasi Artikel Bahasa Indonesia Pendahuluan Studi Terkait," *e-Proceeding Eng.*, vol. 7, no. 2, pp. 8323–8331, 2020.

[4] C. T. Utari, "Implementasi Algoritma Run Length Encoding Untuk Perancangan Aplikasi Kompresi Dan Dekompresi File Citra," *J. TIMES*, vol. V, no. 2, pp. 24–31, 2016, [Online]. Available: http://ejournal.stmik-time.ac.id/index.php/jurnalTIMES/article/download/%0A553/12%0A.

[5] F. N. Pabokory, I. F. Astuti, and A. H. Kridalaksana, "Implementasi Kriptografi Pengamanan Data Pada Pesan Teks, Isi File Dokumen, Dan File Dokumen Menggunakan Algoritma Advanced Encryption Standard," *Inform. Mulawarman J. Ilm. Ilmu Komput.*, vol. 10, no. 1, p. 20, 2016, doi: 10.30872/jim.v10i1.23.

[6] Soumi Rohmah Saragih, Dito Putro Utomo, (2020), "Penarapan Algoritma Prefix Code Dalam Kompresi Data Teks", KOMIK VOL. 4 NO.01

[7] Supiyandi, Okta Frida, 2018, "Analisis Perbandingan Pemanfaatan Data Teks Dengan Menggunakan Metode Humman dan Half-Byte, Jurnal Algoritma Vol.02 No.01 ISSN 2598-6341(Online)

[8] M. Merdiyan and W. Indarto, "Implementasi Algoritma Run Length, Half Byte, dan Huffman untuk Kompresi File," *Semin. Nas. Apl. Teknol. Inf. 2015 (SNATI 2015)*, vol. 2005, no. Snati, pp. 79–84, 2015.