Bulletin of Artificial Intelligence
Volume 1, No 1, April 2022 Page: 33−35
ISSN 2962-3944 (media online)
https://journal.grahamitra.id/index.php/buai

Perbandingan Algoritma Sequitur dan RLE Dalam Kompresi Teks

Rian Syahputra

Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Budi Darma, Medan, Indonesia
Email: ryansyah93@gmail.com

Abstrak–Informasi pada sebuah blog berita tidak sedikit jika kita lihat banyaknya informasi yang akan disampaikan pada sebuah berita. Dalam sebuah blog berita juga menggunakan memori penyimpanan yang tidak sedikit pada database karena banyaknya informasi berupa teks yang disimpan, dan juga mempengaruhi kecepatan transmisi data tersebut. Teknik kompresi yang bertujuan mengurangi data dapat digunakan untuk mengurangi teks yang disimpan dan menghemat media penyimpanan dan mempercepat proses transmisi data. Menggunakan algoritma Sequitur lebih baik dengan rasio kompresi 80.95% dari ukuran awal, dikarenakan algoritma sequitur menghilangkan pasangan symbol yang muncul lebih dari sekali dengan symbol baru. Sedangkan algoritma RLE pada penelitian ini tidak bisa digunakan karena tidak memenuhi algoritma dari RLE tersebut sehingga tidak terjadi proses kompresi dikarenakan tidak adanya karakter yang sama yang muncul berturut-turut, ini merupakan kelemahan dari algoritma RLE.

Keyword: Kompresi Teks; Sequitur; RLE

Abstract–Information on a news blog is not small if we look at the amount of information that will be conveyed in a news story. A news blog also uses a lot of storage memory in the database because of the large amount of information in the form of text that is stored, and also affects the speed of data transmission. Compression techniques aimed at reducing data can be used to reduce stored text and save storage media and speed up the data transmission process. Using the Sequitur algorithm is better with a compression ratio of 80.95% of the initial size, because the sequitur algorithm removes symbol pairs that appear more than once with a new symbol. While the RLE algorithm in this study cannot be used because it does not meet the RLE algorithm so that the compression process does not occur due to the absence of the same characters appearing successively, this is a weakness of the RLE algorithm.

Keyword: Text Compression; Sequiturs; RLE

1. PENDAHULUAN

Banyaknya informasi yang ditulis disebuah blog memungkinkan penggunaan memori pada database yang tidak sedikit. Semakin banyak informasi yang ditulis maka semakin banyak memori yang akan digunakan [1], sehingga akan mempengaruhi besaran memori yang digunakan dan juga waktu loading pada blog tersebut.

Untuk mengurangi penggunaan memori pada database dan waktu loading pada blog, maka bisa dilakukan dengan cara mengurangi isi informasi berita yang disimpan tanpa harus menghilangkan data aslinya saat dimuat pada blog [2]. Salah satu caranya adalah dengan menggunakan Teknik kompresi.

Kompresi digunakan untuk mengurangi Sebagian data pada saat disimpan dan akan dikembalikan pada saat akan digunakan dan Teknik kompresi ini biasanya dipakai untuk data teks [3]. Algoritma yang bisa digunakan untuk teks adalah algoritma Sequitur yang melakukan proses kompresi dengan 2 aturan yaitu Diagram Keunikan (*Diagram Uniqueness*) dan Aturan Kegunaan (*Rule Utility*) [2][3]. Algoritma lain yang digunakan untuk teks adalah Run Length Encoding (RLE) yang melakukan proses kompresi dengan mencari karakter berulang dan diubah menjadi satu karakter dengan byte penanda dengan menghitung jumlah kemunculan karakter dan jumlah dari kemunculan karakter tersebut [4], namun algoritma RLE banyak digunakan untuk kompresi citra dan masih sedikit yang menggunakan untuk kompresi teks [5].

2. METODOLOGI PENELITIAN

2.1 Kompresi Data Teks

Kompresi merupakan proses untuk mengubah data yang bertujuan menghemat memori penyimpanan dan waktu pengiriman data. Proses perubahan data yang terjadi yaitu pada dasarnya dengan menghilangkan perulangan atau *redundancy* dari data asli. Setiap metode kompresi berbeda-beda cara perubahan datanya tetapi tetap sama dalam tujuan [4]. Teks merupakan kumpulan karakter atau string yang bersatu menjadi satu kalimat utuh berupa informasi. Teks dapat menimbulkan masalah jika terdapat banyak karakter yang terkandung di dalamnya, permasalahan yang terjadi biasanya adalah media penyimpanan yang banyak terpakai dan kecepatan dalam transmisi data teks tersebut [1].

2.2 Algoritma Sequitur

Algoritma sequitur melakukan proses kompresi dengan menjalankan Batasan/aturan dalam sebuah tata Bahasa yaitu dengan diagram keunikan (*Diagram Uniqueness*) dan Aturan Kegunaan (*Rule Utility*).

1. Diagram Keunikan memastikan bahwa dalam sebuah tata Bahasa tidak ada pasangan symbol yang muncul lebih dari satu kali, jika ada, maka aturan ini dilanggar dan akan dibentuk aturan baru atau symbol non-terminal yang akan menggantikan symbol tersebut.
2. Aturan kegunaan digunakan untuk memastikan aturan baru yang terbentuk digunakan lebih dari sekali, jika hanya digunakan sekali, maka aturan tersebut harus dihapus [2].

Bulletin of Artificial Intelligence
Volume 1, No 1, April 2022 Page: 33−35
ISSN 2962-3944 (media online)
https://journal.grahamitra.id/index.php/buai

2.3 Algoritma Run Length Encoding (RLE)

Algoritma RLE digunakan dalam proses kompresi yang berisi karakter berulang yang dimana nilai data yang sama pada banyak elemen data yang berturut-turut disimpan sebagai nila data tunggal dan dihitung Panjang elemen tersebut. Teknik kompresi RLE berguna untuk data yang banyak memiliki kesamaan pada deretan data tersebut, sehingga jika data atau elemen yang berdekatan tidak memiliki kesamaan, maka algoritma RLE tidak cocok utnuk digunakan [4].

Contoh sampel: aaaappaaaa kaaaabaaaar?
Dari contoh diatas terdapat 23 karakter dengan ukuran 184 byte, dan dengan proses algoritma RLE maka hasilnya:

Hasil RLE: 4a2p4a1_1k4a1b4a1r1?
Dengan jumlah karakter sebanyak 20 dan ukuran 160 byte.

Dari hasil contoh diatas dapat dilihat bahwa algoritma RLE bekerja dengan melihat karakter berderet yang sama dan diganti nilai data tunggal dan Panjang elemen tersebut [4].

3. HASIL DAN PEMBAHASAN

Untuk mendapatkan hasil yang lebih baik dari kedua metode yang akan dibandingkan, maka akan digunakan sampel data dengan string dari salah satu judul berita "Batasi Konten Jual Beli Organ, Ini Saran Pakar Siber ke Kominfo". Dari sampel diatas kita dapat menghitung bahwa terdapat 63 karakter dan didapat ukuran awal sebesar 504 byte.

3.1 Penerapan Algoritma Sequitur

Ada 2 aturan yang harus dilakukan pada algoritma Sequitur, yaitu diagram keunikan dan aturan kegunaan. Sebelum masuk ke aturan tersebut, terlebih dahulu kita ubah seluruh huruf yang ada menjadi *lowercase* untuk kemudahan dengan hasil sebagai berikut "batasi konten jual beli organ, ini saran pakar siber ke kominfo".

1. Diagram Keunikan

Tabel 1. Diagram Keunikan Algoritma Sequitur

String	Simbol Berulang	Rule	Perubahan String	Simbol non-terminal
batasi konten jual beli organ, ini saran pakar siber ke kominfo	si	si -> A	bataA konten jual beli organ, ini saran pakar Aber ke kominfo	A
bataA konten jual beli organ, ini saran pakar Aber ke kominfo	ko	ko -> B	bataA Bnten jual beli organ, ini saran pakar Aber ke Bminfo	B
bataA Bnten jual beli organ, ini saran pakar Aber ke Bminfo	be	be -> C	bataA Bnten jual Cli organ, ini saran pakar ACr ke Bminfo	C
bataA Bnten jual Cli organ, ini saran pakar ACr ke Bminfo	an	an -> D	bataA Bnten jual Cli orgD, ini sarD pakar ACr ke Bminfo	D
bataA Bnten jual Cli orgD, ini sarD pakar ACr ke Bminfo	in	in -> E	bataA Bnten jual Cli orgD, Ei sarD pakar ACr ke BmEfo	E
bataA Bnten jual Cli orgD, Ei sarD pakar ACr ke BmEfo	ar	ar -> F	bataA Bnten jual Cli orgD, Ei sFD pakF ACr ke BmEfo	F

Dari hasil tabel 1, didapat hasil diagram keunikan dengan string akhir "bataA Bnten jual Cli orgD, Ei sFD pakF ACr ke BmEfo" dengan jumlah karakter 51 dan ukuran 408 byte.

2. Aturan Kegunaan

Dilihat dari hasil diagram keunikan pada tabel 1, maka aturan kegunaan atau *Rule Utility* tidak terjadi pelanggaran sehingga tidak ada symbol yang dihapus atau dihilangkan.

3.2 Penerapan Algoritma Run Length Encoding (RLE)

Algoritma RLE bekerja dengan mencari karakter yang sama dan berderetan dan dihitung jumlah karakter yang sama tersebut. Dari sampel data "Batasi Konten Jual Beli Organ, Ini Saran Pakar Siber ke Kominfo" akan dicari karakter yang sama berturut-turut dan disimpan dalam bentuk nilai tunggal dan dihitung panjangnya.

Tabel 2. Algoritma RLE

String	Perubahan String
batasi konten jual beli organ, ini saran pakar siber ke kominfo	-

Dilihat dari tabel 2, bahwasannya string sampel tersebut tidak terdapat karakter yang sama berturut-turut sehingga tidak terjadi proses kompresi dengan algoritma RLE, dan ukuran masih tidak ada perubahan.

3.3 Perbandingan Hasil Algoritma

Dari penerapan kedua algoritma dapat dilihat hasil analisis sebagai berikut:

1. Hasil algoritma Sequitur

Rasio kompresi $= \frac{ukuran\ setelah\ dikompresi}{ukuran\ sebelum\ dikompresi}$ x 100%

$= \frac{408}{504}$ x 100% = 80.95%

2. Hasil algoritma RLE

Tidak ada rasio karena tidak memenuhi aturan dari algoritma RLE sehingga tidak terjadi proses kompresi.

4. KESIMPULAN

Dari hasil penelitian dapat di simpulan bahwa algoritma Sequitur lebih baik digunakan untuk teks daripada algoritma RLE dengan rasio 80.95% daripada ukuran awal, dikarenakan algoritma Sequitur memiliki aturan diagram keunikan yang memastikan tidak ada pasangan symbol yang muncul lebih dari sekali, jika melanggar maka akan diganti dengan symbol non-terminal yang baru. Algoritma RLE sangat tidak cocok jika digunakan untuk kompresi teks, karena sangat jarang teks atau string memiliki karakter atau elemen yang berderetan atau berturut-turut memiliki kesamaan karena data teks memiliki pola yang tersusun sehingga dapat menjadi sebuah kata yang memiliki arti. Algoritma RLE lebih cocok digunakan untuk kompresi file karena memiliki kemunculan elemen yang sama berturut-turut lebih tinggi daripada teks.

REFERENCES

[1] S. R. Saragih and D. P. Utomo, "Penarapan Algoritma Prefix Code Dalam Kompresi Data Teks," *KOMIK (Konferensi Nas. ...*, vol. 4, no. 1, pp. 249–252, 2020, doi: 10.30865/komik.v4i1.2691.

[2] Y. Darnita, K. Khairunnisyah, and H. Mubarak, "Kompresi Data Teks Dengan Menggunakan Algoritma Sequitur," *Sistemasi*, vol. 8, no. 1, p. 104, 2019, doi: 10.32520/stmsi.v8i1.429.

[3] S. Siahaan, "Penerapan Algoritma Sequitur Pada Kompresi Record Database Pada Database," *JURIKOM (Jurnal Ris. Komputer)*, vol. 6, no. 5, pp. 511–516, 2019, [Online]. Available: https://www.ejurnal.stmik-budidarma.ac.id/index.php/jurikom/article/view/1644

[4] U. Mansyuri, "KOMPRESI DATA TEKS DENGAN METODE RUN LENGTH ENCODING," *J. SIMASI*, vol. 1, no. 2, pp. 102–109, 2021.

[5] Herdianto, "Perbandingan Metode RLE dan Huffman dalam Kompresi Data Teks," *J. Ilm. Core It*, vol. 9, no. 1, pp. 41–47, 2019.