



JURNAL SIMASI

p-ISSN 2798-835X (Print) e-ISSN 2798-8341 (Online)

Journal Page is available to <http://simasi.lppmbinabangsa.id/index.php/home>



KOMPRESI DATA TEKS DENGAN METODE RUN LENGTH ENCODING

¹Umar Mansyuri

¹Universitas Bina Bangsa, Kota Serang-Banten

Email: umar.mansyuri@gmail.com

ABSTRAK

Salah satu metode penggunaan kompresi data yaitu dengan menggunakan metode yang disebut dengan *Run Length Encoding (RLE)*, khususnya data citra. Metode RLE merupakan salah satu jenis lossless yang paling sederhana dari skema kompresi data dan didasarkan pada prinsip sederhana data encoding.

Metode RLE sangat cocok untuk mengompresi data yang berisi karakter-karakter berulang misalnya citra grafis sederhana. Data yang dikompresi adalah 28 citra RGB (Red, Green, Blue) dan 28 citra grayscale masing-masing dalam format jpg, png, bmp, dan tiff. Data citra dikompresi dengan program encoder dan decoder menggunakan algoritma RLE pada aplikasi matlab.

Metode RLE dikatakan efektif mengompresi data citra jika rasio kompresi kurang dari 100% dikarenakan memiliki banyak perulangan warna pada piksel-pikselnya. Metode RLE dikatakan tidak efektif jika rasio kompresi lebih dari 100% dikarenakan memiliki sedikit perulangan warna pada piksel-pikselnya. Dari 28 citra RGB yang diujicoba, dihasilkan bahwa metode RLE efektif pada 1 citra dan tidak efektif pada 27 citra. Untuk 28 citra grayscale yang diujicoba, dihasilkan bahwa metode RLE efektif pada 6 citra dan tidak efektif pada 22 citra.

Kata kunci: Run Length Encoding, kompresi, piksel

ABSTRACT

One method of using data compression is by using a method called Run Length Encoding (RLE), especially image data. The RLE method is one of the simplest lossless types of data compression schemes and is based on the simple principle of data encoding.

The RLE method is very suitable for compressing data containing repetitive characters such as simple graphic images. The compressed data are 28 RGB (Red, Green, Blue) images and 28 grayscale images in jpg, png, bmp, and tiff formats, respectively. Image data is compressed with an encoder and decoder program using the RLE algorithm in the matlab application.

The RLE method is said to be effective in compressing image data if the compression ratio is less than 100% because it has a lot of color repetition in the pixels. The RLE method is said to be ineffective if the compression ratio is more than 100% because it has a little repetition of colors in the pixels. Of the 28 RGB images tested, it was found that the RLE method was effective on 1 image and not effective on 27 images. For the 28 grayscale images tested, it was found that the RLE method was effective on 6 images and not effective on 22 images

Keyword: Run Length Encoding, compression, pixels

PENDAHULUAN

Semakin berkembangnya teknologi dan berkembang pesatnya arah perkembangan zaman dalam teknologi dalam hal penyimpanan data membuat para vendor berlomba-lomba mengoptimalkan kapasitas data ditambah dengan semakin pesatnya arus data yang dalam pengiriman yang semakin besar sehingga dibutuhkan penghematan bandwidth, sehingga mengoptimalkan pengiriman data yang mengakibatkan pengiriman data akan semakin lebih cepat.

Dan hal tersebut membuat parah ahli menciptakan sesuatu dalam hal penghematan data atau pengecilan data (kompresi data) membuat algoritma atau metode kompresi data. Dalam kompresi data setiap data memiliki bit bit merangkai suatu file data, dan dari bit tersebut akan di manipulasi dikurangi nilainya tapi tidak menghilangkan data-datanya.

Ada dua tipe utama kompresi data, yaitu kompresi tipe lossless dan kompresi tipe lossy. Kompresi tipe lossy adalah kompresi dimana terdapat data yang hilang selama proses kompres. Akibatnya kualitas citra yang dihasilkan jauh lebih rendah dari pada kualitas citra asli. Sementara itu, kompresi tipe lossless tidak menghilangkan informasi setelah proses kompresi terjadi, akibatnya kualitas citra hasil kompresi tidak menurun. Ada banyak sekali teori dan metode untuk kompresi data, di antaranya metode *Entropy Encoding* (Huffman), *Run-Length Encoding* (RLE), *Lempel-Zip-Welch* (LZW), *Shanon-Fano*, dan beberapa metode lainnya

LANDASAN TEORI

2.1 Kompresi Data Teks

1. **Teks** adalah satuan lingual yang dimediasi secara tulis atau lisan dengan tata organisasi tertentu untuk mengungkapkan makna secara kontekstual. Dalam kompresi data teks terdiri dari karakter karakter yang menjadikan suatu kesatuan.
2. **Kompresi** adalah suatu teknik atau cara pemampatan data sehingga hanya memerlukan penyimpanan data yang kecil dan mengefisiensikan data sehingga mempersingkat waktu pengiriman data
3. **Kompresi Data Teks** adalah Proses pemampatan data berupa teks sehingga mengurangi jumlah data dalam teks. Contoh kompresi sederhana misalnya kata “yang “dikompresi menjadi “yg” Pengiriman data hasil 1 hasil kompresi dapat di lakukan jika pihak pengirim dan pihak penerima memiliki aturan yang sama dalam hal kompresi data.

Kompresi data menjadi sangat penting karena memperkecil kebutuhan penyimpanan data, mempercepat pengiriman data, memperkecil kebutuhan bandwidth

2.2 Pendekatan Kompresi

Ada empat pendekatan yang digunakan pada kompresi suatu data, yaitu:

1. **Pendekatan statistic**

Kompresi didasarkan pada frekuensi kemunculan derajat keabuan pixel didalam seluruh bagian.

Contoh metode: Huffman Coding.

2. Pendekatan ruang

Kompresi didasarkan pada hubungan spasial antara pixel-pixel di dalam suatu kelompok yang memiliki derajat keabuan yang sama dalam suatu daerah gambar atau data.

Contoh metode: Run-Length Encoding.

3. Pendekatan kuantisasi

Kompresi dilakukan dengan mengurangi jumlah derajat keabuan yang tersedia.

Contoh metode: kompresi kuantisasi (CS&Q).

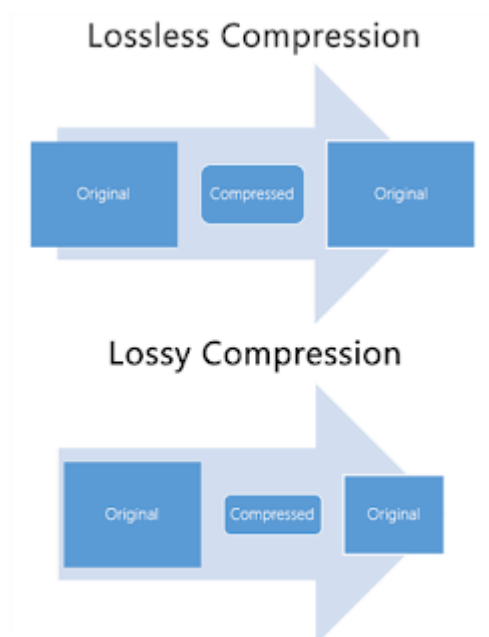
4. Pendekatan fractal

Kompresi dilakukan pada kenyataan bahwa kemiripan bagian-bagian didalam data atau citra atau gambar dapat dieksploitasi dengan suatu matriks transformasi.

Contoh metode: Fractal Image Compression.

2.3 Metode Kompresi

Pada kompresi data terdapat 2 jenis kompresi yaitu lossless dan lossy. berikut merupakan gambar perbedaan lossless dan lossy:



Metode Lossless dan lossy akan dijelaskan sebagai berikut:

1. Metode lossless

Lossless data kompresi adalah kelas dari algoritma data kompresi yang memungkinkan data yang asli dapat disusun kembali dari data kompresi. Lossless data kompresi digunakan dalam berbagai aplikasi seperti format ZIP dan GZIP. Lossless juga sering digunakan sebagai komponen dalam teknologi kompresi data lossy. Kompresi Lossless digunakan ketika sesuatu yang penting pada kondisi asli. Beberapa format gambar seperti PNG atau GIF hanya menggunakan kompresi lossless, sedangkan yang lainnya seperti TIFF dan MNG dapat menggunakan metode lossy atau lossless.

Metode lossless menghasilkan data yang identik dengan data aslinya, hal ini dibutuhkan untuk banyak tipe data, contohnya: executable code, word processing files, tabulated numbers, dan sebagainya. Misalnya pada citra atau gambar dimana metode ini akan menghasilkan hasil yang tepat sama dengan citra semula, pixel per pixel sehingga tidak ada informasi yang hilang akibat kompresi. Namun ratio kompresi (Rasio kompresi yaitu, ukuran file yang dikompresi dibanding yang tak terkompresi dari file) dengan metode ini sangat rendah. Metode ini cocok untuk kompresi citra yang mengandung informasi penting yang tidak boleh rusak akibat kompresi, misalnya gambar hasil diagnosa medis. Contoh metode lossless adalah metode run-length, Huffman, delta dan LZW. TEKNIK KOMPRESI LOSSLESS Kebanyakan program kompresi lossless menggunakan dua jenis algoritma yang berbeda: yang satu menghasilkan model statistik untuk input data, dan yang lainnya memetakan data input ke rangkaian bit menggunakan model ini dengan cara bahwa “probable” data akan menghasilkan output yang lebih pendek dari “improbable” data. algoritma encoding yang utama yang dipakai untuk menghasilkan rangkaian bit adalah Huffman coding dan Aritmatik Coding.

2. Metode lossy

Lossy kompresi 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). Sebaliknya, 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 di level bit ketika tidak dapat dibedakan oleh mata dan telinga manusia untuk tujuan paling praktis.

Metode ini menghasilkan ratio kompresi yang lebih besar daripada metode lossless. Misal terdapat image asli berukuran 12,249 bytes, kemudian dilakukan kompresi dengan JPEG kualitas 30 dan berukuran 1,869 bytes berarti image tersebut 85% lebih kecil dan ratio kompresi 15%. Contoh metode lossy adalah metode CS&Q (coarser sampling and/or quantization), JPEG, dan MPEG.

Ada dua skema dasar lossy kompresi:

- **Lossy transform codec**, sampel suara atau gambar yang diambil, di potong kesegmen kecil, diubah menjadi ruang basis yang baru, dan kuantisasi. hasil nilai kuantisasi menjadi entropycoded
- **Lossy predictive codec**, sebelum dan/atau sesudahnya data di-decode digunakan untuk memprediksi sampel suara dan frame picture saat ini. kesalahan antara data prediksi dan data yang nyata, bersama-sama dengan informasi lain digunakan untuk mereproduksi prediksi, dan kemudian dikuantisasi dan kode.

2.4 RLE (RUN LENGTH ENCODING)

Kompresi data teks yang dilakukan jika ada huruf yang sama yang ditampilkan berturut-turut. Kompresi dan dekompresi data menggunakan teknik *Run length encoding* ini merupakan suatu bentuk teknik yang digunakan untuk mengkompresi data yang berisi karakter-karakter berulang (Hauck, S. & Wilson, William D., 1999: *IEEE*). Saat karakter yang sama diterima secara berderet empat kali atau lebih (lebih dari tiga), algoritma ini mengkompres data dalam suatu tiga karakter berderetan.

Teknik yang digunakan pada algoritma ini adalah dengan mencari karakter yang berulang lebih dari 3 kali pada suatu file untuk kemudian diubah menjadi sebuah bit penanda (*marker bit*) diikuti oleh sebuah bit yang memberikan informasi jumlah karakter yang berulang dan kemudian ditutup dengan karakter yang dikompres, yang dimaksud dengan bit penanda di sini adalah deretan 8-bit yang membentuk suatu karakter ASCII.

HASIL DAN PEMBAHASAN

3.1 Metode RLE (Run Length Encoding)

Pada pembahasan ini akan di bahas mengenai kompresi data teks dengan menggunakan salah satu metode kompresi data yaitu metode Run Length Encoding (RLE). Kompresi dan dekompresi data menggunakan *run-length encoding* (RLE) ini merupakan suatu bentuk teknik yang digunakan untuk mengkompresi data yang berisi karakter-karakter berulang.

Run-length encoding (RLE) adalah bentuk yang sangat sederhana dari kompresi data di mana data berjalan (yaitu, urutan di mana nilai data yang sama terjadi pada banyak elemen data yang berturut-turut) disimpan sebagai nilai data tunggal dan dihitung panjangnya. Hal ini berguna pada data yang berisi banyak data berjalan, misalnya: gambar grafis sederhana seperti ikon, gambar garis, dan animasi. Hal ini tidak berguna dengan file yang tidak memiliki banyak data berjalan karena sangat dapat meningkatkan ukuran file.

3.2 Proses Penerapan Metode RLE

kompresi data teks dilakukan jika terdapat beberapa huruf yang sama ditampilkan secara berturut-turut. Terdapat dua tipe RLE yaitu RLE tipe 1 dan RLE tipe 2.

Contoh:

Data: ABCCCCCCCCDEFGGGG = 17 karakter

Dengan RLE tipe 1 (min. 4 huruf sama) ditulis; ABC8! DEFG!4 = 11 karakter

Dalam RLE tipe 1 ini terdapat suatu karakter yang tidak digunakan dalam teks seperti tanda ‘!’ yang digunakan untuk menandai. Teknik kompresi RLE tipe 1 ini memiliki kelemahan yaitu jika terdapat karakter angka, mana tanda mulai dan tanda akhir? Maka dalam RLE tipe 2 digunakanlah flag bilangan negatif untuk menandai batas sebanyak jumlah karakter tersebut.

Contoh:

Data: ABCCCCCCCCDEFGGGG = 17 Karakter

Dengan RLE tipe 2: -2AB8CDEF4G = 12 Karakter

Contoh:

Data; AB12CCCCDEEEF = 13 Karakter

Dengan RLE tipe 2; -4AB124CD3EF = 12 Karakter

Teknik kompresi dengan RLE ini berguna untuk data yang banyak memiliki kesamaan dan data tersebut berdekatan, misal teks ataupun grafik seperti icon atau gambar garis-garis yang banyak memiliki kesamaan pola.

Misalkan, ada seseorang yang berteriak:

“AAAAKUUUUU SAHAAAAABBBBBBAT
KAAAAMUUUUUUUUUUUUUUUUUUUUUUUUUUUUU !!!!!!”

Pesan tersebut akan sangat cocok jika dikompresi menggunakan metode kompresi RLE karena kompresi RLE menghitung jumlah kemunculan simbol lalu menuliskan simbol tersebut

sebanyak satu kali diikuti dengan jumlah kemunculannya. Data diatas berukuran 66 byte, dan kita akan melakukan kompresi RLE terhadap data tersebut :

➤ **Ubah data dalam bentuk sekuensial**

Data teks diatas sudah dalam bentuk sekuensial :

AAAAAKUUUUU SAHAAAAABBBBBBAT
KAAAAMUUUUUUUUUUUUUUUUUUUUUUUUUUUU!!!!!!

➤ **Hitung jumlah kemunculan karakter**

(A,6) (K,1) (U,5) (spasi,1) (S,1) (A,1) (H,1) (A,5) (B,5) (A,1) (T,1) (spasi,1) (K,1) (A,5)
(M,1) (U,24) (!,6)

➤ **Tulis hasil kompresi**

A6K1U5 1C1H1A1Y5A5N1K1 1K1A5M1U24!6

Setelah proses kompresi, maka data yang dihasilkan akan berukuran 35 byte. Dengan proses kompresi tersebut, kita telah menghemat tempat penyimpanan sebesar 31 byte (47%).

Algoritma metode kompresi RLE hanya efisien dengan data *file* yang berisi kelompok data (*byte* / karakter) yang berulang dan dapat digunakan pada *file* teks. *File* teks berisi banyak kelompok data yang berupa spasi atau tabulator, tetapi juga dapat diterapkan untuk citra (gambar) yang berisi area hitam atau putih yang besar.

KESIMPULAN

Dari pembahasan dan dasar teory di atas dapat disimpulkan :

- *Run-length encoding* (RLE) adalah bentuk yang sangat sederhana dari kompresi data
- kompresi RLE menghitung jumlah kemunculan simbol lalu menuliskan simbol tersebut sebanyak satu kali diikuti dengan jumlah kemunculannya
- Teknik kompresi dengan RLE ini berguna untuk data yang banyak memiliki kesamaan dan data tersebut berdekatan, misal teks ataupun grafik seperti icon atau gambar garis-garis yang banyak memilki kesamaan pola.

DAFTAR PUSTAKA

- Utari, C. T. (2016). Implementasi Algoritma Run Length Encoding Untuk Perancangan aplikasi Kompresi Dan Dekompresi File Citra. *Jurnal Times*, 5(2), 24-31.
- Pujianto, P., Mujito, M., Prasetyo, B. H., & Prabowo, D. (2020). Perbandingan Metode Huffman dan Run Length Encoding Pada Kompresi Dokumen. *InfoTekJar: Jurnal Nasional Informatika dan Teknologi Jaringan*, 5(1), 216-223.

- Prayoga, E., & Suryaningrum, K. M. (2018). Implementasi Algoritma Huffman Dan Run Length Encoding Pada Aplikasi Kompresi Berbasis Web. *Jurnal Ilmiah Teknologi Infomasi Terapan*, 4(2).
- Lu'luilmaknun, U., & Salsabila, N. H. (2017). Penggunaan Metode Run Length Encoding Untuk Kompresi Data. In *Prosiding Seminar Matematika dan Pendidikan Matematika UNY 2017*.
- Sriwiyanto, A. (2012). Implementasi Metode Run Length Encoding Dalam Kompresi Citra Dengan Citra Hitam Putih (Doctoral dissertation, Universitas Muhammadiyah Surakarta).