



Penyempurnaan Algoritma Run-Length Encoding Menggunakan Algoritma Burrows-Wheeler Transform Pada Pemampatan Data Teks

Didik Trisianto¹

¹Fakultas Ilmu Komputer, Universitas Narotama, Surabaya, Indonesia, didik.trisianto@narotama.ac.id

STATUS ARTIKEL

Dikirim 15 April 2025
Direvisi 25 April 2025
Diterima 28 April 2025

Kata Kunci:

Kompresi, Algoritma, Analisis, Teks

ABSTRAK

Penelitian ini membahas tentang kinerja algoritma Run-Length Encoding dengan peningkatan algoritma Burrows-Wheeler Transform pada kompresi data. Ruang lingkup penelitian ini hanya dibatasi pada kompresi data teks (*.txt) yang dilakukan dua kali percobaan yaitu percobaan dengan menggunakan file teks yang berasal dari potongan artikel dan percobaan dengan menggunakan file teks dengan satu variasi karakter. Kriteria yang digunakan dalam perbandingan hasil percobaan adalah rasio kompresi dan analisis menggunakan ANOVA. Pada percobaan potongan artikel tersebut menunjukkan bahwa rasio kompresi yang dihasilkan oleh algoritma Run-Length Encoding dengan menggunakan algoritma Burrows-Wheeler Transform kemungkinan lebih baik dibandingkan dengan algoritma yang dilakukan dengan Run-Length Encoding saja. Untuk rasio kompresi pada data percobaan menunjukkan hasil yang homogen tidak berbeda nyata antara penggunaan Run-Length Encoding dan itu adalah sebesar 90,83%. Hal ini dikarenakan kompresi Run-Length Encoding bekerja paling baik pada file yang memiliki karakter dengan potensi kemunculan mendekati 1 (hampir sama dengan bobot ukuran file). Rasio kompresi Run-Length Encoding terburuk terjadi pada file yang memiliki karakter beragam ditandai dengan terjadinya pembesaran ukuran file dibanding ukuran file dari awal.

1. PENDAHULUAN

Majunya peradaban membuat manusia terus berusaha untuk meningkatkan kualitas komunikasi jarak jauh. Untuk itu, informasi sebagai komponen utama dari komunikasi perlu disajikan dengan sebaik-baiknya. Terdapat tiga metode utama dalam menyajikan informasi: melalui ucapan, tulisan, dan representasi grafis. Informasi yang direpresentasikan dalam bentuk tulisan umumnya dikenal sebagai data teks.

Sebelum teknologi informasi berkembang, data biasa disimpan dan disajikan dalam media kertas atau yang lebih dikenal dengan media konvensional. Penyimpanan menggunakan media konvensional kurang menguntungkan karena data dengan jumlah besar akan memerlukan kertas dan ruang penyimpanan yang besar pula. Pada media konvensional penyimpanan data tidak begitu baik serta membutuhkan waktu yang lama untuk melakukan pengiriman. Perkembangan teknologi membuat media komputer dijadikan sebagai media penyimpanan dan pengiriman data. Data yang disimpan menggunakan media komputer biasa disebut dengan data digital.

Pada data digital dikenal istilah pemampatan. Pemampatan adalah suatu metode untuk memperkecil ukuran data dari ukuran semula. Dengan pemampatan, kebutuhan media penyimpanan menjadi lebih efisien. Selain berguna untuk penyimpanan data, kompresi data dapat membantu memperkecil ukuran data yang ditransmisikan di dalam suatu media jaringan, seperti internet. Dengan ukuran data yang lebih kecil, maka waktu yang diperlukan untuk mentransfer data tersebut dapat menjadi lebih kecil.

Ukuran data teks tidak sebesar ukuran data gambar apabila disimpan dalam suatu media digital. Namun untuk kepentingan efisiensi ruang penyimpanan serta mempercepat waktu transmisi maka dianggap perlu untuk melakukan suatu proses pemampatan pada data teks. Pemampatan data teks bertujuan untuk mengurangi pengulangan penggunaan simbol atau karakter yang tersusun dengan cara mengkodekan simbol-simbol atau karakter tersebut.

Salah satu teknik kompresi data yang mapan adalah Run-Length Encoding (RLE), yang beroperasi dengan prinsip reduksi string berulang. Oleh sebab itu, RLE menunjukkan kinerja optimal pada citra digital dengan keragaman karakter yang rendah. Perkembangan terkini menghadirkan Algoritma Burrows-Wheeler Transform (BWT), yang berfungsi untuk mereorganisasi karakter dalam data sehingga lebih kondusif untuk proses kompresi menggunakan algoritma Run-Length Encoding.

2. METODE PENELITIAN

Rasio pemampatan algoritma Run-Length Encoding tergantung pada perubahan karakter dari string, sehingga algoritma ini tidak dapat bekerja secara optimal pada data dengan perubahan karakter yang sangat bervariasi. Saat ini algoritma Run-Length Encoding lebih sering digunakan untuk kompresi image gambar dimana variasi perubahan karakter dalam image gambar tidak terlampau besar.

Algoritma Burrows-Wheeler Transform adalah algoritma tanpa hak paten yang melakukan tugas pengolahan string sehingga menjadi bentuk yang siap untuk diolah lebih lanjut menggunakan algoritma Run-Length Encoding. Penelitian ini berusaha untuk menganalisis kinerja algoritma Run-Length Encoding pada kompresi data teks yang telah disempurnakan menggunakan algoritma Burrows-Wheeler Transform.

Hipotesis

Berdasarkan studi literatur, ada beberapa hipotesis yang akan dicoba untuk dibuktikan.

1. Pada data heterogen

Pemampatan data menggunakan algoritma Run-Length Encoding (RLE) pada data heterogen diprediksi tidak efektif atau bahkan dapat menghasilkan ukuran file terkompresi yang lebih besar daripada ukuran file aslinya. Hal ini didasarkan pada prinsip kerja RLE yang hanya melakukan reduksi ukuran data apabila terdapat karakter-karakter berdekatan dengan nilai yang identik. Pada data heterogen, variasi karakter yang tinggi dan minimnya pengulangan karakter berdekatan akan menggagalkan mekanisme kompresi RLE, bahkan berpotensi menambah informasi overhead yang menyebabkan pembengkakan ukuran file.

2. Pada data homogen

Terdapat perbedaan signifikan dalam rasio kompresi yang dihasilkan antara penggunaan algoritma Run-Length Encoding (RLE) dasar dengan algoritma RLE yang disempurnakan pada data homogen.

Analisis Kinerja Algoritma

Hasil percobaan akan dianalisis dengan menggunakan kriteria-kriteria berikut :

1. Kebutuhan tempat penyimpanan

Seluruh file yang diuji perlu dicatat ukuran file semula dan ukuran file setelah dikompresi. Hal ini diperlukan untuk mendapatkan rasio pemampatan yang dilakukan oleh algoritma tersebut. Perhitungan rasio pemampatan didefinisikan dengan rumus (Ruth dan Krentzler, 1972):

$$\text{Rasio Pemampatan} = \frac{\text{loss}}{\text{ukuran file asli}}$$

dimana-

loss = ukuran file awal – ukuran file kompresi
(ukuran file yang hilang akibat ter-mampatkan)

2. Penarikan kesimpulan menggunakan analisis keragaman (ANOVA).

Bahan

Percobaan dilakukan dengan menggunakan dua program pemampatan data teks yang dibuat oleh Mark Nelson, di download dari internet (<http://dogma.net/markn/articles>), yaitu :

1. Program pemampatan menggunakan RLE
2. Program transformasi BWT.

Kedua program tersebut terdiri dari proses pengkodean dan pendekodean, yang dibangun dengan menggunakan bahasa pemrograman C. BWT dan RLE dibuat dalam dua program yang terpisah, sehingga dapat dipilih penggunaannya. Perincian perintah yang digunakan adalah sebagai berikut :

1. RLE < file_asal | RLE > file_hasil, jika hanya RLE yang akan digunakan untuk melakukan kompresi.
2. RLE < file_asal | BWT | RLE > file_hasil, jika kompresi diawali dengan BWT baru kemudian dilanjutkan oleh RLE.

Data

Data yang digunakan adalah data teks dengan ekstensi *.txt dengan berbagai rentang ukuran file. Berbagai rentang ukuran file yang digunakan yaitu < 20.000 byte, antara 20.000 sampai 40.000 byte dan file dengan ukuran > 40.000 byte. Data tersebut dibagi menjadi dua kategori.

1. Data heterogen, data ini diperoleh dari file teks berisi kutipan tajuk berencana harian sebuah media cetak online (Media Indonesia) yang beredar antara bulan Juni sampai November 2003.
2. Data homogen, diperoleh dengan menggunakan sebuah mekanisme untuk membangkitkan karakter dalam sebuah file.

Pembagian data menjadi dua kategori ini berguna untuk mempermudah pembuktian hipotesis. Sedangkan pembagian data menjadi beberapa rentang bertujuan untuk melihat rata-rata peningkatan rasio pada peningkatan ukuran file yang dikompresi.

Struktur Percobaan

Penelitian ini dibagi menjadi dua tahap :

1. Percobaan pada data heterogen (artikel)
Tiap rentang file (yaitu < 20.000 byte, antara 20.000 sampai 40.000 byte dan file > 40.000 byte), akan dikompresi menggunakan RLE dan RLE+BWT.
2. Percobaan pada data homogen
Tiap rentang file (yaitu < 20.000 byte, antara 20.000 sampai 40.000 byte dan file > 40.000 byte), akan dikompresi menggunakan RLE dan RLE+BWT.

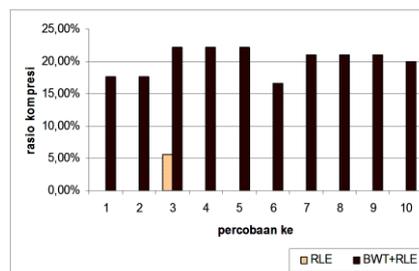
3. HASIL DAN PEMBAHASAN

Hasil iterasi kedua algoritma pada saat ditampilkan akan dibedakan dengan warna, pada hasil iterasi algoritma Run-Length Encoding ditunjukkan dengan warna abu muda dan untuk hasil penyempurnaan algoritma Run-Length Encoding menggunakan algoritma Burrows-Wheeler Transform ditunjukkan dengan warna abu tua.

A. Data Heterogen

A.1. Untuk rentang di bawah 20.000 byte

Dari hasil percobaan dapat dilihat bahwa secara umum kompresi menggunakan RLE yang disempurnakan dengan BWT menghasilkan rasio kompresi yang jauh lebih baik daripada hanya menggunakan algoritma RLE. Pada rentang ini, rata-rata rasio kompresi menggunakan RLE dan BWT adalah 20,18%, sedangkan penggunaan RLE menghasilkan rata-rata rasio kompresi sebesar 0,56%. Grafik perbandingan rasio untuk rentang ini dapat dilihat pada Gambar 1.

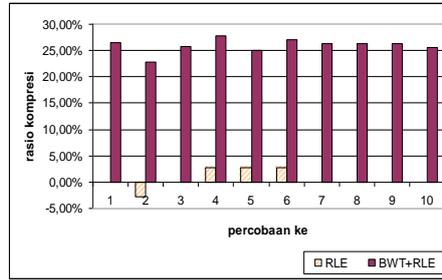


Gambar 1. Perbandingan besar rasio kompresi kedua algoritma pada rentang file < 20.000 byte.

A.2. Untuk rentang 20.000 sampai 40.000 byte

Untuk rentang ini juga dapat dilihat bahwa secara umum kompresi menggunakan RLE yang disempurnakan dengan BWT menghasilkan rasio kompresi yang jauh lebih baik daripada hanya menggunakan algoritma RLE. Bahkan pada file ke-2 rasio kompresi bernilai minus. Nilai minus menunjukkan kasus dimana file hasil kompresi mempunyai ukuran yang lebih besar daripada file asal. Pada rentang ini, rata-rata rasio kompresi

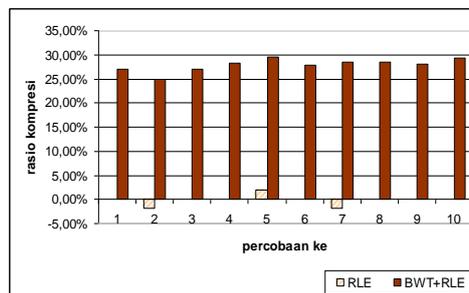
menggunakan RLE dan BWT adalah 25,94%, sedangkan penggunaan RLE menghasilkan rata-rata rasio kompresi sebesar 0,54%. Grafik perbandingan rasio untuk rentang ini dapat dilihat pada Gambar 2.



Gambar 2. Perbandingan besar rasio kompresi kedua algoritma pada rentang antara 20.000 sampai 40.000 byte.

A.3. Untuk rentang di atas 40.000 byte

Gambar 7 menunjukkan grafik hasil percobaan pada rentang file dengan ukuran di atas 40.000 byte. Untuk rentang ini dapat dilihat bahwa secara umum kompresi menggunakan RLE yang disempurnakan dengan BWT menghasilkan rasio kompresi yang jauh lebih baik daripada hanya menggunakan algoritma RLE. Pada percobaan ke-2 dan ke-7 rasio kompresi bernilai minus. Hal ini menunjukkan hal dimana pada kasus tersebut file hasil kompresi mempunyai ukuran yang lebih besar daripada file asal. Pada rentang ini, rata-rata rasio kompresi menggunakan RLE dan BWT adalah 27,906%, sedangkan penggunaan RLE menghasilkan rata-rata rasio kompresi sebesar 0,19%. ?

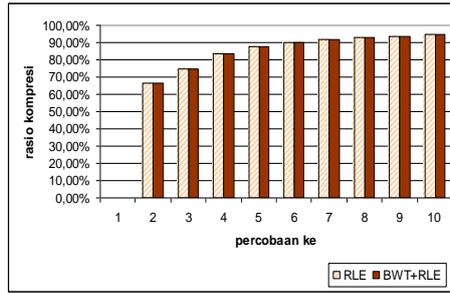


Gambar 3. Perbandingan besar rasio kompresi kedua algoritma pada rentang file > 40.000 byte.

B. Data Homogen

A.1. Untuk rentang di bawah 20.000 byte

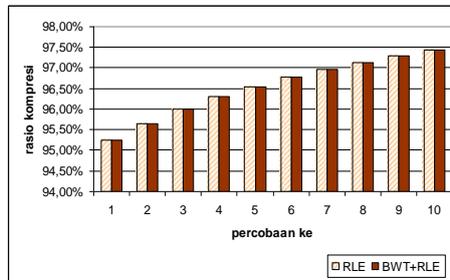
Dari percobaan didapatkan hasil bahwa rasio kompresi baik dari kompresi RLE maupun penyempurnaannya mempunyai nilai yang sama. Hasil perbandingannya dapat dilihat pada Gambar 4. Rata-rata rasio kompresi pada rentang ini adalah 77,52%.



Gambar 4. Perbandingan besar rasio kompresi kedua algoritma pada rentang file < 20.000 byte.

A.2. Untuk rentang 20.000 sampai 40.000 byte

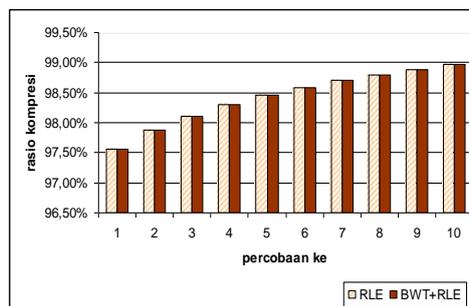
Gambar 5 menunjukkan hasil kompresi pada file sumber berukuran antara 20.000 byte sampai dengan 40.000 byte. Sama seperti rentang yang sebelumnya, proses kompresi pada dua perlakuan menghasilkan rasio yang sama. Rata-rata rasio kompresi pada rentang ini adalah 96,54%.



Gambar 5. Perbandingan besar rasio kompresi kedua algoritma pada rentang antara 20.000 sampai 40.000 byte.

A.3. Untuk rentang di atas 40.000 byte

Tidak berbeda dengan rentang-rentang yang sebelumnya, perlakuan pada file-file berukuran di atas 40.000 byte juga menghasilkan rasio kompresi yang sama. Namun rata-rata rasio kompresinya meningkat menjadi 98,42%. Grafik yang menyajikan hasilnya dapat dilihat pada Gambar 6.



Gambar 6. Perbandingan besar rasio kompresi kedua algoritma pada rentang file > 40.000 byte.

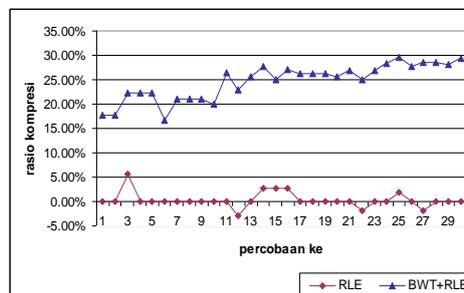
PEMBAHASAN

A. Data Heterogen

Ada dua hipotesis yang akan dibuktikan pada rentang percobaan ini. Yang pertama adalah rasio kompresi yang tidak terlalu baik. Dan yang kedua adalah hasil file kompresi yang bisa jadi lebih besar ukurannya dibandingkan file asal.

Gambar 7 menyajikan hasil percobaan yang telah dilakukan dalam satu grafik. Hipotesis pertama dapat dibuktikan dengan melihat hasil percobaan. Rata-rata rasio kompresi menggunakan algoritma RLE pada data heterogen adalah 0,30% dengan perincian : 0,56% pada tentang file < 20.000 byte, 0,54% untuk rentang file antara 20.000 sampai 40.000 byte, dan -0,19% untuk rentang file berukuran > 40.000 byte. Nilai rasio ini tidak terlalu baik jika dibandingkan dengan besarnya rasio kompresi jika menggunakan algoritma gabungan RLE dan BWT yaitu sebesar 24,67%.

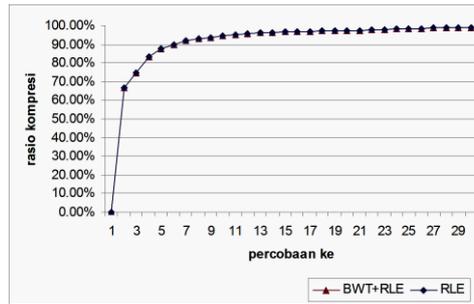
Hipotesis kedua terbukti dengan adanya hasil rasio kompresi negatif pada percobaan ke-12, 22, 27 dengan nilai masing-masing -2,86%, -1,92%, dan -1,78%. Hasil tersebut mengindikasikan diperolehnya file hasil kompresi yang lebih besar daripada file asal. Penyebab terjadinya rasio negatif ini adalah sedikitnya karakter yang sama pada letak yang berdekatan dalam kondisi atau pola dalam data yang menjadi target utama untuk dikompresi menggunakan metode seperti Run-Length Encoding. Seperti telah dibahas pada teori, algoritma RLE melakukan kompresi dengan memampatkan karakter-karakter yang sama pada letak yang berdekatan. Jika tidak ada karakter yang sama pada letak yang berdekatan maka algoritma menjadi tidak efektif karena yang dilakukan oleh adalah menambah banyaknya karakter yang disimpan.



Gambar 7. Perbandingan besar rasio kompresi pada data heterogen menggunakan RLE dan BWT+RLE.

B. Data Homogen

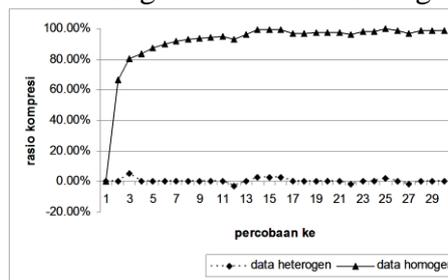
Sesuai dengan hipotesis, pada data homogen tidak ada perbedaan antara penggunaan algoritma RLE yang baik disempurnakan dengan BWT maupun tidak. Hasil ini dapat dilihat pada Gambar 8, besar rata-rata kompresi pada data homogen ini adalah 90,83%.



Gambar 8. Perbandingan besar rasio kompresi pada data homogen menggunakan RLE dan BWT+RLE.

C. Kinerja algoritma RLE

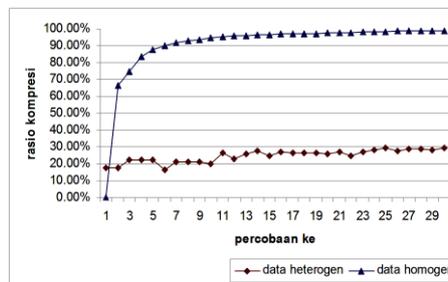
Algoritma RLE bekerja dengan melakukan reduksi pada string-string yang berulang. Rasio pemampatan algoritma Run-Length Encoding tergantung pada perubahan karakter dari string, sehingga algoritma ini tidak dapat bekerja secara optimal pada data dengan perubahan karakter yang sangat bervariasi. Saat ini algoritma Run-Length Encoding lebih sering digunakan untuk kompresi image gambar dimana variasi perubahan karakter dalam image gambar tidak terlampau besar. Gambar 9 merepresentasikan hal dimana RLE sangat efektif bekerja pada data homogen namun tidak terlalu bagus untuk data heterogen.



Gambar 9. Perbandingan besar rasio kompresi menggunakan algoritma RLE pada data homogen dan heterogen.

C. Kinerja algoritma RLE + BWT

Gambar 10 merepresentasikan peningkatan rasio kompresi jika digunakan algoritma BWT sebagai pelengkap. Namun, rasio kompresi pada data homogen tetap menghasilkan rasio yang lebih baik daripada data heterogen.



Gambar 10. Perbandingan besar rasio kompresi menggunakan algoritma RLE + BWT pada data homogen dan heterogen.

D. Pengujian Menggunakan ANOVA

Tujuan dari pengujian ini adalah untuk membantu penarikan kesimpulan akhir dari data-data hasil percobaan dengan menggunakan pendekatan ragam. Kriteria yang diuji adalah rasio kompresi. Hasil pengujian menggunakan analisis ANOVA menghasilkan beberapa hal.

Pertama, pada data heterogen dihasilkan P-value dari dua faktor yang ada (rentang dan algoritma) dengan nilai masing-masing $<0.0001\%$. Ini berarti bahwa baik perlakuan beda rentang ukuran maupun beda algoritma menghasilkan rasio kompresi yang berbeda nyata. Atau dengan kata lain bisa ditarik kesimpulan bahwa perlakuan menggunakan BWT+RLE berbeda nyata dengan perlakuan menggunakan RLE, dan percobaan juga menghasilkan rasio kompresi yang berbeda nyata antara tiap rentang yang ditetapkan (<20.000 byte, $20.000-40.000$, dan > 40.000 byte). Hasil analisis ANOVA untuk data heterogen disajikan pada Tabel 1.

Tabel 1. Hasil ANOVA pada data heterogen

S. Keragaman	dF	MS	F _{hitung}	P-value
Rentang (R)	2	0.00695002	27.24	<.0001
Algoritma (A)	1	0.89094283	3492.02	<.0001
Interaksi R & A	2	0.00936019	36.69	<.0001
Error (E)	54	0.00025514		
Total (T)	59			

Kemudian untuk menganalisis dimana letak perbedaan nyata, dilakukan analisis lanjut menggunakan Duncan Grouping. Pada analisis Duncan Grouping, perlakuan dengan hasil yang berbeda nyata ditunjukkan dengan penulisan karakter yang berbeda. Tabel 2 merepresentasikan Duncan Grouping pada faktor 1 (besar file). Pada analisis ini diperoleh hasil yang berbeda nyata pada perlakuan file <20.000 byte dibandingkan perlakuan yang lainnya.

Tabel 2. Duncan Grouping pada faktor ke-1 data heterogen

Duncan Grouping	Mean	N	faktor1
A	0.138605	20	>40
A	0.132400	20	20-40
B	0.103665	20	<20

Sedangkan pada Tabel 3 disajikan Duncan Grouping pada faktor 2 (algoritma). Pada analisis ini diperoleh hasil yang berbeda nyata antara perlakuan menggunakan RLE dan menggunakan RLE+BWT.

Tabel 3. Duncan Grouping pada faktor ke-2 data heterogen

Duncan Grouping	Mean	N	faktor2
A	0.246747	30	RLE+BWT
B	0.003033	30	RLE

Selanjutnya, pada data yang memiliki varians homogen, analisis statistik menunjukkan bahwa variasi pada rentang perlakuan menghasilkan perbedaan yang signifikan secara statistik ($p = 0.0002$). Sebaliknya, tidak ditemukan adanya perbedaan signifikan dalam hasil yang disebabkan oleh penggunaan algoritma yang berbeda ($p = 1.0000$). Ini berarti bahwa perlakuan beda rentang ukuran mengasilkan beda nyata. Namun perlakuan beda algoritma tidak mengasilkan rasio kompresi yang berbeda nyata. Hasil analisis ANOVA untuk data heterogen disajikan pada Tabel 4. Hasil analisis ini diperjelas dengan melakukan analisis lanjutan menggunakan Duncan Grouping.

Tabel 4. Hasil ANOVA pada data homogen

S. Keragaman	dF	MS	F _{hitung}	P-value
Rentang (R)	2	0.26733011	9.73	0.0002
Algoritma (A)	1	0.00000000	0.00	1.0000
Interaksi R & A	2	0.00000000	0.00	1.0000
Error (E)	54	0.02746113		
Total (T)	59			

Untuk faktor 1 (perbedaan ukuran), diperoleh hasil yang berbeda nyata pada perlakuan file <20.000 byte dibandingkan perlakuan yang lainnya. Tabel 5 merepresentasikan Duncan Grouping untuk faktor 1.

Tabel 5. Duncan Grouping pada faktor ke-1 data homogen

Duncan Grouping	Mean	N	faktor1
A	0.98424	20	>40
A	0.96536	20	20-40
B	0.77522	20	<20

Sesuai hasil analisis menggunakan ANOVA, tidak ditemukan perbedaan nyata antara penggunaan algoritma yang berbeda (baik RLE maupun RLE+BWT), hal ini dibuktikan dengan hasil karakter yang sama (A) pada hasil analisis Duncan Grouping di Tabel 6.

Tabel 6. Duncan Grouping pada faktor ke-2 data heterogen

Duncan Grouping	Mean	N	faktor2
A	0.90827	30	RLE
A	0.90827	30	RLE+BWT

4. KESIMPULAN

Algoritma RLE bekerja dengan melakukan reduksi pada string-string yang berulang. Rasio pemampatan algoritma Run-Length Encoding tergantung pada perubahan karakter dari string, sehingga algoritma ini tidak dapat bekerja secara optimal pada data dengan perubahan karakter yang sangat bervariasi.

Rasio kompresi optimal pada RLE dicapai ketika tingkat redundansi dalam file sangat tinggi akibat dominasi satu atau beberapa karakter yang muncul secara berurutan dengan frekuensi yang sangat tinggi. Dalam kasus ekstrem, bobot karakter yang berulang ini mendekati total ukuran file. Rasio terbaik yang didapat selama penelitian yaitu sebesar 98.97%.

Rasio kompresi terburuk RLE terjadi pada file yang memiliki variasi karakter besar dan pada file yang berukuran kecil. Hal ini ditandai dengan terjadinya pembesaran ukuran file hasil dibandingkan ukuran file awal. Hal ini dibuktikan pada percobaan ke-2 rentang file yang kedua dimana hasil kompresi menghasilkan nilai sebesar -2.86%.

Kompleksitas algoritma RLE adalah $O(N)$, sedangkan kompleksitas algoritma BWT (Nelson, 1996) adalah $O(N * \log(N))$.

BWT melakukan penyusunan kembali terhadap karakter-karakter dalam string berdasarkan kesamaan nilainya, sehingga menghasilkan blok-blok string dengan nilai yang sama. BWT membuat rasio kompresi RLE menjadi lebih baik, hal ini dibuktikan pada percobaan dimana penggunaan BWT menaikkan rasio kompresi pada data heterogen sebesar 24.37% dengan perincian 20.178%, 25.940%, 27.906% pada tiap rentang file.

Algoritma BWT melakukan pengelompokan karakter pada string per-blok dalam suatu file. Sehingga algoritma ini tidak sepenuhnya melakukan pengurutan karakter dalam file tersebut. Oleh karena itu, pengembangan penelitian lebih lanjut dapat dilakukan dengan menyempurnakan algoritma RLE + BWT pada kompresi data teks menggunakan algoritma penyempurna yang lain seperti Move to Front. Penelitian ini masih mempunyai kelemahan dalam hal analisis waktu, penghitungan waktu kompresi dilakukan dengan melakukan estimasi berdasarkan kompleksitas algoritma. Sehingga pada penelitian lebih lanjut diharapkan dapat dilakukan analisis dari sudut pandang waktu kompresi.

5. DAFTAR PUSTAKA

- Astrigina, M. 2001. Perbandingan Metode Huffman dan Lempel-Ziv-Welch untuk Pemampatan Berkas Teks [skripsi]. Bogor: Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Pertanian Bogor.
- Campos, A. Run Length Encoding, [http:// www.arturocampos.com/ac_rle.html](http://www.arturocampos.com/ac_rle.html) [5 desember 2004].
- Dewi, W.N. 2003. Pemampatan File dengan Menggunakan Metode LZSS [skripsi]. Bogor: Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Pertanian Bogor.

- Held, G. 1943. Data Compression: Techniques and Application Hardware and Software Considerations. Ed ke-3. West Sussex : John Wiley & Sons Ltd.
- Matjjik, A.A. 2000. Perancangan Percobaan. Ed ke-2. Bogor: IPB Press.
- Moore, David S. 1994. The Basic Practice of Statistics. ISBN 0-7167-2628-9. New York : W.H. Freeman and Company.
- Nelson, M. Data Compression with the Burrows-Wheeler Transform. <http://dogma.net/markn/articles/bwt/bwt.htm>. [17 Desember 2003].
- Sulistio, D.D. 2003. Perbandingan Algoritma Huffman Statik Dengan Algoritma Huffman Adaptif Pada Kompresi Data Teks[skripsi]. Bogor: Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Pertanian Bogor.
- Yusran, R. 2004. Implementasi dan Analisis Efisiensi Beberapa Algoritma Kompresi Teks Terhadap Aspek-Aspek Informasi Tekstual [skripsi]. Depok: Fakultas Ilmu Komputer Universitas Indonesia.