

Aplikasi Algoritma *Greedy* dalam Kompresi Dokumen Teks

Nurul Fithria Lubis 13510012¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

¹13510012@std.stei.itb.ac.id

Abstrak— Algoritma *greedy* ialah salah satu jenis algoritma yang sering digunakan dalam penyelesaian masalah. Dalam algoritma *greedy*, untuk setiap langkah penyelesaian masalah akan diambil solusi optimum dengan harapan pada akhir penyelesaian didapatkan pula hasil optimum. Di samping itu, di tengah perkembangan teknologi, suatu isu yang menjadi perhatian ialah penggunaan memori yang mangkus. Melalui algoritma *greedy*, penghematan itu dapat dilakukan, misalnya dalam kompresi dokumen teks. Algoritma *greedy* akan berperan dalam menentukan kata mana yang dapat diolah sehingga ukuran teks semakin kecil. Pemilihan ini dapat dilakukan dengan mempertimbangkan dua aspek, yaitu panjang kata dan frekuensi kata. Namun, kedua jenis *greedy* yang diuji memberikan hasil akhir yang relatif sama dengan beberapa perbedaan insignifikans secara tekstual. Algoritma *greedy* berhasil memangkas ukuran teks dengan baik, namun ada beberapa kekurangan pada aplikasinya yang membutuhkan pengembangan lebih lanjut.

Kata kunci— dokumen, *greedy*, kompresi, teks.

I. PENDAHULUAN

Kompresi adalah hal yang secara sadar maupun tidak sadar dilakukan manusia setiap hari. Menggulung kabel, melipat baju, menumpuk gelas, hal ini dilakukan demi ruang yang lebih luas untuk kebutuhan lainnya. Dalam kegiatan sehari-hari, kompresi menjadi hal yang krusial pada saat bepergian karena ruang barang yang sangat terbatas untuk barang-barang yang akan dibawa. Untuk menyisati hal itu, banyak industri yang membuat barang-barang inovatif seperti kemasan kedap udara untuk memperkecil ruang suatu barang. Selain itu, banyak produk dibuat dengan kemasan kecil khusus untuk kebutuhan bepergian.

Demikian pula halnya dengan data yang disimpan pada komputer. Tanpa kompresi sama sekali, kebutuhan *disk storage* dari seorang pengguna komputer akan menjadi sangat besar. Walaupun harga memori semakin murah, namun kebutuhan penyimpanan dari pengguna juga semakin besar. Maka dari itu, tentu hal ini tetap perlu disiasati agar pemakaian memori dilakukan semangkas mungkin.

Teknik kompresi data sudah berkembang pesat dari

zaman ke zaman. Setiap jenis file membutuhkan teknik kompresi yang berbeda pula karena strukturnya yang sejatinya berbeda. Sebagai contoh, kompresi pada gambar dapat dilakukan dengan JPEG dan GIF, pada audio dapat dilakukan dengan MP3, dan pada video dapat dilakukan dengan MP4.

Jenis file lain yang tidak dapat dipungkiri juga membutuhkan kompresi ialah teks. Pada awalnya, pengolahan dokumen tidak menjadi masalah karena ukurannya relatif kecil. Namun, seiring dengan berjalannya waktu dan bertambahnya kebutuhan pengolahan data, timbul permasalahan mengenai penggunaan memori pada saat pengelolaan dan penyimpanan dokumen karena ukurannya bisa menjadi sangat besar.

Banyak teknik kompresi yang telah dikembangkan untuk kompresi dokumen teks, namun pada makalah ini penulis akan membahas penggunaan algoritma *greedy* untuk menyelesaikan permasalahan tersebut.

II. DASAR TEORI

A. Algoritma *Greedy*

Algoritma *greedy* ialah salah satu tipe algoritma yang membentuk solusi per langkah dari sebuah masalah. Dari setiap langkah yang dilakukan, mungkin terdapat banyak pilihan jalan yang dapat diambil untuk mencapai tujuan akhir. Strategi dari algoritma *greedy* ialah mengambil suatu keputusan yang terbaik dalam setiap langkah dari pemecahan masalah. Setiap keputusan yang telah diambil dengan *greedy* pada setiap langkah bersifat final dan tidak dapat diubah.

Pendekatan dari algoritma *greedy* adalah mengambil pilihan yang memberikan perolehan terbaik pada saat itu, hal ini biasa disebut dengan istilah optimum lokal. Pada setiap pengambilan, diharapkan pengambilan optimum lokal ini akan menuju pada solusi optimum global.

Jika disimpulkan, algoritma *greedy* adalah algoritma yang menyelesaikan masalah secara langkah per langkah dengan mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan dan berharap optimum lokal akan menuju pada

optimum global.

Terdapat lima komponen *greedy* sebagai berikut:

1. Himpunan kandidat, *C*, berisi elemen-elemen pembentuk solusi. Pada setiap langkah *greedy*, satu elemen akan dipilih dari sini untuk mendapatkan optimum lokal.
2. Himpunan solusi, *S* yang berisi kandidat solusi permasalahan
3. Fungsi seleksi, Fungsi yang memilih di antara sekian banyak elemen dalam kandidat dan mengembalikan elemen kandidat yang paling optimum.
4. Fungsi kelayakan yang memeriksa apakah kandidat yang dipilih dari fungsi seleksi memenuhi persyaratan dan tidak melanggar aturan.
5. Fungsi obyektif, fungsi yang meminimumkan / memaksimalkan suatu solusi.

Greedy bersifat “*pick what you can get now*”, maka dari itu *greedy* tidak akan selalu memberikan solusi yang optimum global. Pada kondisi seperti ini, *greedy* menjadi basis pendekatan heuristik.

Walaupun tidak selalu memberikan solusi optimum global, algoritma *greedy* tetap menjadi salah satu algoritma yang sering dipilih karena kecepatan eksekusinya. Dan seringkali walaupun solusi yang diberikan bukanlah solusi optimum global, solusi tersebut sudah sangat baik.

B. Kompresi Data

Kompresi data adalah 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 bertujuan untuk merepresentasikan informasi pada suatu file digital dengan sesedikit mungkin bit. Untuk itu, perlu dilakukan beberapa langkah pengolahan data melalui algoritma yang didefinisikan.

Algoritma untuk kompresi data memiliki dua jenis, yakni algoritma *lossy* dan *lossless*.

Sesuai dengan namanya, algoritma *lossy* menyebabkan hilangnya sebagian data pada proses kompresinya. Namun, pada algoritma *lossless*, proses kompresi dapat di-reverse (dekomposisi) sehingga data dapat kembali seperti semula yang berarti tidak ada bagian data yang hilang.

Namun, pada algoritma kompresi *lossy*, hilangnya sebagian data tidak akan mengubah secara drastis arti dari dokumen karena pemangkasan tersebut sudah dilakukan sedemikian rupa sehingga walaupun tanpa dekomposisi, pengguna tetap bisa memanfaatkan dokumen yang sudah dikompresi.

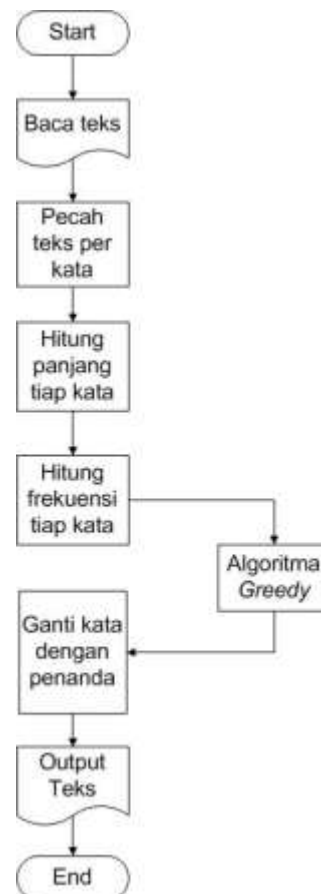
C. Dokumen Teks

Dokumen teks ialah dokumen dengan tujuan utama baca dan tulis. Terdapat beberapa format dokumen teks,

pada tulisan ini dokumen teks yang dimaksud ialah dokumen dengan ekstensi .txt dan format ASCII 8 bit. Pada dokumen ini, setiap katakter diberikan ukuran sebesar 1byte atau 8 bit.

III. METODE PENYELESAIAN MASALAH

Kompresi terhadap teks akan dilakukan dengan basis kata. Teks akan dipecah menjadi kata perkata, untuk setiap kata, disimpan tiga informasi, yakni kata itu sendiri, frekuensi, serta jumlah karakternya. Alur kerja dari penyelesaian masalah tergambar pada Gambar 1.



Gambar 1 Alur kerja kompresi data

Sebagai contoh, akan digunakan teks berikut:

Jingle bells, jingle bells, jingle all the way
of the bells bells.

Pemecahan teks akan menghasilkan tabel1.

Tabel 1 Hasil pemecahan teks

Kata	Jumlah karakter	Frekuensi
jingle	6	3/12
bells	5	4/12
all	3	1/12
the	3	2/12
way	3	1/12
of	2	1/12

Tabel inilah yang akan digunakan dalam penentuan skema *greedy*. Dalam tulisan ini akan dibahas dua jenis algoritma *greedy* dalam kompresi, yakni *greedy by length* dan *greedy by value*. Kedua jenis algoritma *greedy* ini diuji untuk memberi perbandingan, apakah salah satu lebih baik dari yang lain atau apakah ada *trade-off* pada keduanya.

Kata yang dipilih oleh algoritma *greedy* kemudian akan digantikan dengan penanda yang tentunya akan dibuat sependek mungkin agar ukuran teks berkurang. Namun, penanda ini tidak boleh merupakan karakter atau kumpulan karakter yang sering muncul pada teks karena dapat menyebabkan kesalahpahaman.

A. Greedy by Length

Dengan menggunakan teks contoh sebagai masalah, pada algoritma *greedy by length*, skema algoritma *greedy* didefinisikan sebagai berikut:

1. Himpunan kandidat (C)
Seluruh kata yang ada pada Tabel 1
2. Fungsi Seleksi
Anggota himpunan C dengan jumlah karakter paling sedikit..
3. Fungsi Kelayakan
Jumlah karakter dari anggota C yang dipilih lebih besar dari jumlah karakter dari penanda yang digunakan.
4. Fungsi Obyektif
Anggota C yang dipilih memiliki jumlah karakter terkecil dibanding anggota himpunan C lainnya dan jumlah karakternya lebih besar dari jumlah karakter penanda.
5. Himpunan Solusi (S)
Kata yang memenuhi fungsi obyektif

Iterasi pertama pada teks dengan skema *greedy* di atas menghasilkan Tabel 2. Kata yang menjadi anggota S ialah *jingle* karena dibandingkan kata lainnya, *jingle* memiliki jumlah karakter terbesar. Sebuah tanda akan diberikan untuk kata *jingle*.

Tabel 2 Hasil iterasi pertama dengan *greedy by length*

Kata	Jumlah karakter	Frekuensi	Penanda
jingle	6	3/12	!a
bells	5	4/12	-
all	3	1/12	-
the	3	2/12	-
way	3	1/12	-
of	2	1/12	-

Lalu, dilakukan penandaan pada kata *jingle*, sehingga setelah iterasi pertama, didapatkan teks sebagai berikut:

!a bells, !a bells, !a all the way of the bells bells.

Iterasi kedua pada teks akan menghasilkan Tabel 3.

Kata berikutnya yang menjadi anggota himpunan solusi S ialah *bells* karena dibanding kata-kata lainnya, *bells* memiliki jumlah karakter terbesar. Kata *jingle* tidak lagi masuk ke dalam daftar kata karena sudah digantikan dengan penanda !a sehingga tidak lagi memenuhi fungsi kelayakan dan fungsi obyektif.

Seperti pada iterasi pertama, diberikan sebuah tanda untuk kata *bells*. Tanda ini akan menggantikan seluruh kata *bells* pada teks.

Tabel 3 Hasil iterasi kedua dengan *greedy by length*

Kata	Jumlah karakter	Frekuensi	Penanda
bells	5	4/12	!b
all	3	1/12	-
the	3	2/12	-
way	3	1/12	-
of	2	1/12	-
!a	2	3/12	-

Dan seterusnya, iterasi akan terus dilakukan hingga himpunan solusi S pada algoritma *greedy* memberikan himpunan kosong. Pada akhirnya, didapatkan teks berikut:

!a !b, !a !b, !a !c !d !e of !d !b !b.

B. Greedy by Frequency

Dengan menggunakan teks contoh sebagai masalah, pada algoritma *greedy by length*, skema algoritma *greedy* didefinisikan sebagai berikut:

1. Himpunan kandidat (C)
Seluruh kata yang ada pada Tabel 1
2. Fungsi Seleksi
Anggota himpunan C dengan frekuensi paling besar.
3. Fungsi Kelayakan
Jumlah karakter dari anggota C yang dipilih lebih besar dari jumlah karakter dari penanda dan frekuensinya lebih besar dari satu.
4. Fungsi Obyektif
Anggota C yang dipilih memiliki frekuensi terbesar dibanding anggota himpunan C lainnya dan jumlah karakternya lebih besar dari jumlah karakter penanda.
5. Himpunan Solusi (S)
Kata yang memenuhi fungsi obyektif

Iterasi pertama pada teks dengan skema *greedy* di atas menghasilkan Tabel 4. Kata yang menjadi anggota S ialah *bells* karena dibandingkan kata lainnya, *bells* memiliki frekuensi terbesar. Sebuah tanda akan diberikan untuk kata *bells*.

Tabel 4 Hasil iterasi pertama dengan greedy by frequency

Kata	Jumlah karakter	Frekuensi	Penanda
jingle	6	3/12	-
bells	5	4/12	!a
all	3	1/12	-
the	3	2/12	-
way	3	1/12	-
of	2	1/12	-

Lalu, dilakukan penandaan pada kata bells, sehingga setelah iterasi pertama, didapatkan teks sebagai berikut:

```
jingle !a, jingle !a, jingle all the way of the !a !a.
```

Iterasi kedua pada teks akan menghasilkan Tabel 5. Kata berikutnya yang menjadi anggota himpunan solusi S ialah jingle karena dibanding kata-kata lainnya, jingle memiliki frekuensi terbesar. Kata bells tidak lagi masuk ke dalam himpunan solusi S karena seluruhnya sudah digantikan dengan penanda !a sehingga tidak lagi memenuhi fungsi kelayakan dan fungsi obyektif.

Seperti pada iterasi pertama, diberikan sebuah tanda untuk kata jingle. Tanda ini akan menggantikan seluruh kata jingle pada teks.

Tabel 5 Hasil iterasi kedua dengan greedy by length

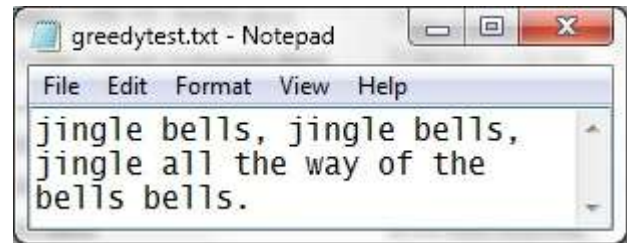
Kata	Jumlah karakter	Frekuensi	Penanda
jingle	6	3/12	!b
all	3	1/12	-
the	3	2/12	-
way	3	1/12	-
of	2	1/12	-
!a	2	4/12	-

Dan seterusnya, iterasi akan terus dilakukan hingga himpunan solusi S memberikan himpunan kosong. Sehingga pada akhirnya didapatkan teks berikut:

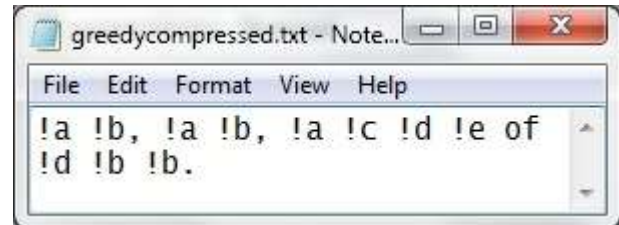
```
!b !a, !b !a, !b !d !c !e of !c !a !a.
```

IV. PEMBAHASAN

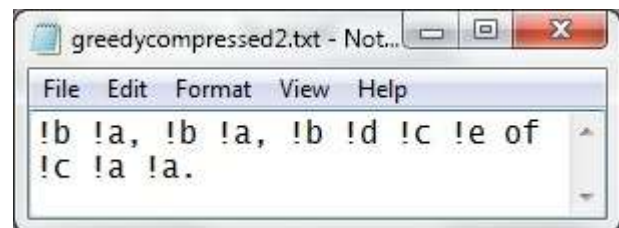
Dengan menggunakan teks pada contoh, dilakukan kompresi *greedy by length* dan juga *greedy by frequency*. Untuk mengetahui hasil sebenarnya dari kompresi, teks disimpan dalam dokumen teks kemudian dilihat dan dibandingkan ukurannya. Teks sebelum dan sesudah kompresi dapat dilihat pada Gambar 2, Gambar 3, dan Gambar 4.



Gambar 2 Dokumen teks sebelum kompresi

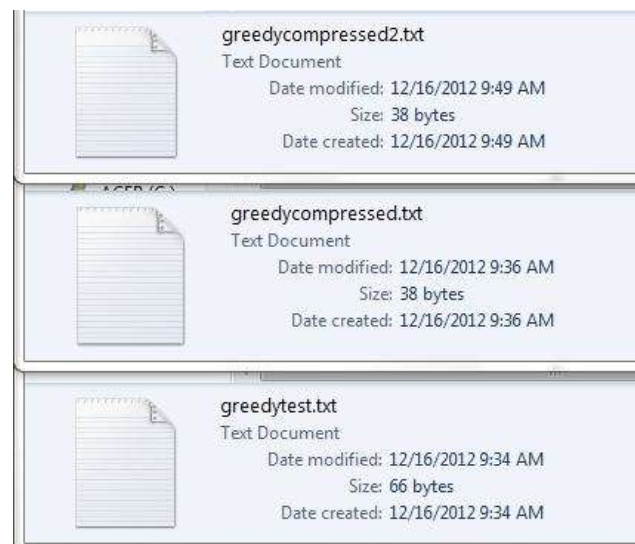


Gambar 3 Dokumen teks setelah kompresi greedy by length



Gambar 4 Dokumen teks setelah kompresi greedy by frequency

Setelah kompresi dilakukan, dokumen teks disimpan dan dilihat ukurannya. Perbandingan ukuran dari ketiga dokumen teks dapat dilihat pada Gambar 5.



Gambar 5 Perbandingan ukuran dokumen sebelum dan sesudah kompresi

Terlihat perbedaan yang cukup signifikan dari ukuran dokumen teks sebelum dan sesudah kompresi, di mana ukuran dokumen teks berkurang hampir setengahnya untuk kedua jenis kompresi, dari 66bytes menjadi 38 bytes. Hal ini terjadi karena berkurangnya panjang teks,

sehingga berkurang pula ukuran dokumen teksnya karena untuk setiap karakter, dialokasikan 1 byte.

Namun, pada kedua jenis kompresi tampak tidak ada perbedaan signifikan. Kedua jenis *greedy* memberikan ukuran akhir dokumen sebesar 38 bytes. Secara tekstual, terdapat perbedaan pada teks hasil kompresi. Pada *greedy by length*, !a menandakan jingle karena jingle adalah kata dengan karakter terbanyak, sedangkan pada *greedy by frequency* !a menandakan bells karena bells adalah kata dengan frekuensi terbesar.

Namun, hal ini tidak berpengaruh pada ukuran akhir dari dokumen teks karena panjang penanda yang digunakan untuk setiap kata sama. Selain itu, kata-kata yang diubah menjadi tanda pada kedua jenis *greedy* juga sama karena fungsi kelayakan yang dari keduanya juga sama.

Maka dari itu, mungkin perlu dilakukan analisis lebih lanjut terhadap fungsi kelayakan yang paling tepat pada algoritma *greedy* yang digunakan dan juga terhadap pemilihan penanda yang digunakan sebagai pengganti kata.

V. KEKURANGAN DAN BATASAN

Aplikasi algoritma *greedy* untuk kompresi pada dokumen teks yang dibahas pada tulisan ini memiliki beberapa kekurangan dalam praktiknya.

Lingkup bahasan masih sangat sempit, di mana eksperimen hanya dilakukan pada dokumen dengan ekstensi .txt saja. Di samping itu, varian kata yang digunakan masih sedikit dan teks yang digunakan pendek sehingga masalah yang harus diselesaikan masih sangat kecil. Karena itu, kemanjuran dari teknik kompresi ini masih belum teruji untuk permasalahan yang lebih besar.

Penanda kata yang digunakan pada contoh masih sangat terbatas, karena dengan tanda !x, dengan x adalah alfabet dari a-z, hanya terdapat 26 tanda yang dapat digunakan dan tentu hal itu menjadi batasan pada kompresi teks yang panjang dengan varian kata yang sangat beragam. Namun di lain sisi, jika jumlah karakter penanda diperpanjang untuk memperbanyak kemungkinan tanda untuk memperluas jangkauan kompresi, kompresi akan menjadi kurang mangkus. Maka dari itu, diperlukan analisa lebih lanjut dalam penentuan penanda kata yang paling tepat untuk digunakan.

Selain itu, untuk teks agar tetap menjadi berarti untuk pengguna, informasi penanda juga harus disimpan agar bisa dilakukan dekompresi. Hal ini tidak diinginkan karena membutuhkan memori lagi untuk penyimpanan informasi tersebut.

VI. KESIMPULAN

Algoritma *greedy* dapat digunakan sebagai algoritma dalam teknik kompresi dokumen teks berbasis kata. Awalnya teks akan dipisah menjadi kata per kata untuk kemudian dipilih kata yang akan digantikan dengan penanda. Penanda harus didefinisikan dahulu sebelumnya

karena akan menentukan fungsi kelayakan dari algoritma *greedy*.

Kompresi data ini bersifat *lossless* karena dekompresi dapat dilakukan melalui informasi arti penanda. Namun hal ini kurang diinginkan karena dibutuhkan memori lagi untuk menyimpan informasi tersebut. Selain itu, perlu dilakukan pengembangan lebih lanjut dalam penentuan penanda dan fungsi kelayakan dari algoritma *greedy* yang paling baik untuk digunakan.

Namun, secara keseluruhan algoritma *greedy* dapat digunakan dalam teknik kompresi dokumen teks.

REFERENCES

- [1] Munir, Rinaldi. (2009). Diktat Kuliah IF3051 Strategi Algoritma. Bandung: Program Studi Teknik Informatika, Institut Teknologi Bandung.
- [2] NN. Digital Collection
<http://digilib.petra.ac.id/viewer.php?submit.x=9&submit.y=13&page=2&qual=high&submitval=prev&fname=%2Fjunkspe%2Fs1%2Finfo%2F2009%2Fjunkspe-ns-s1-2009-26404111-11745-deflate-chapter2.pdf>
Akses terakhir : 16 Desember 2012, 12:25.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 17 Desember 2012



Nurul Fithria Lubis
13510012