

# Perbandingan Kompresi Data Menggunakan Algoritma Huffman dan Algoritma DMC

Emil Fahmi Yakhya - 13509069  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganessa 10 Bandung 40132, Indonesia  
emilfahmi@students.itb.ac.id

**Abstrak**—Makalah ini meninjau dua buah algoritma kompresi yang banyak digunakan namun memiliki karakteristik yang berbeda. Kedua algoritma tersebut adalah algoritma Huffman dan algoritma DMC (Dynamic Markov Compression). Kedua algoritma tersebut tergolong ke dalam jenis yang sama namun memiliki karakteristik pemrosesan yang berbeda. Kedua algoritma tersebut tergolong ke dalam jenis algoritma *lossless compression scheme*, yakni pemrosesan kompresi data menggunakan kedua algoritma tersebut tidak akan memberikan kehilangan data sehingga kedua algoritma tersebut cocok untuk digunakan dalam aplikasi peringkasan ukuran file. Pada algoritma Huffman, kita akan melihat karakteristik statik pada prosesnya. Seluruh data dalam bentuk karakter akan dididat satu persatu sebelum dilakukan pembentukan pohon Huffman untuk enkripsi secara rekursif. Sedangkan pada algoritma DMC, kita akan mengenal proses yang dinamik dimana proses mengambil satu karakter sebagai inisial state dan kemudian dilakukan proses *next state* dan *input-output* untuk transisi antar state secara dinamis. Perbedaan karakteristik ini akan memberikan efek yang cukup signifikan yang memiliki keunggulan dan kekurangan masing-masing.

**Kata kunci:** Algoritma, Huffman, DMC, Statik, Dinamik

## 1. PENDAHULUAN

Di era informasi sekarang ini, proses penciptaan serta pengiriman data dari satu tempat ke tempat lain berjalan dengan sangat cepat. Segala proses tersebut dijalankan tanpa henti selama 24 jam setiap hari.

Dalam perjalanannya, kebutuhan untuk melakukan transfer data dalam jarak jauh ataupun melalui suatu media penyimpanan menjadi semakin tinggi. Namun, di sini terdapat sebuah permasalahan yakni dengan bertambahnya kebutuhan untuk melakukan perpindahan data tersebut, media penyimpanan yang dapat diciptakan untuk menunjang prosesnya tidak selalu bertambah dengan kecepatan yang sebanding untuk proses tersebut.

Oleh karena itu, lahirlah sebuah pemikiran yang didasari kebutuhan penghematan memori yang digunakan untuk menyimpan data. Segala proses untuk penghematan dan peringkasan memori yang digunakan dalam komputer ini kita kenal dengan nama *Data Compressing*, atau lebih dikenal dengan kompresi data.

Dalam pengertian ilmu komputer, kompresi data adalah proses pengkodean informasi yang menggunakan bit (atau unit penyimpanan informasi lainnya) lebih sedikit dari penggunaan bit yang seharusnya digunakan sebelum kompresi. Secara umum terdapat dua jenis algoritma kompresi yakni *lossless data compression* dan *lossy data compression*. Kedua jenis algoritma kompresi data tersebut memiliki karakteristik yang cukup kontras, yakni perbedaan dalam keutuhan data yang dikompresi setelah *encoding*.

### 1.1 Algoritma Huffman

Algoritma Huffman diciptakan oleh seorang mahasiswa MIT bernama David Huffman pada tahun 1952. Algoritma Huffman merupakan metode yang paling tua dan paling terkenal dalam kompresi, khususnya teks. Algoritma Huffman termasuk ke dalam jenis algoritma yang menggunakan *lossless compression scheme* atau dalam bahasa Indonesianya adalah skema kompresi tanpa kehilangan. Istilah tersebut bermakna bahwa algoritma Huffman melakukan proses kompresi tanpa menghilangkan bit yang digunakan untuk menyimpan memori sedikitpun. Cara kerja algoritma ini mirip dengan kode morse yakni setiap karakter (simbol) dikodekan hanya dengan rangkaian beberapa bit dimana karakter yang paling sering muncul dikodekan dengan rangkaian bit yang pendek dan karakter yang jarang muncul dikodekan dengan rangkaian bit yang lebih panjang.

### 1.2 Algoritma DMC

Algoritma DMC adalah algoritma yang memanfaatkan teknik pemodelan yang didasarkan pada model *finite-state* (*proses Markov*). Algoritma ini dikembangkan oleh Gordon Cormack dan Horspool Nigel. Algoritma DMC menggunakan *predictive arithmetic coding* yang mirip dengan PPM (Prediction by Partial Matching) sehingga memungkinkan kemampuan kompresi file yang tinggi. Algoritma ini merupakan jenis algoritma kompresi data yang sama dengan algoritma Huffman. Dan algoritma ini juga merupakan algoritma yang termasuk populer digunakan.

## 2. DASAR TEORI

## 2.1 Algoritma Huffman

Algoritma Huffman merupakan jenis algoritma kompresi data tanpa menghilangkan memori objek. Algoritma ini digunakan untuk melakukan kompresi pada file-file yang berukuran besar menjadi file yang berukuran lebih kecil. Implementasi algoritma Huffman banyak digunakan untuk melakukan kompresi file yang bertipe string. Proses pada algoritma Huffman digolongkan sebagai pemrosesan statik.

Langkah-langkah penggunaan algoritma Huffman dalam kompresi file string dapat dijabarkan sebagai berikut :

1. Tahap Seleksi
  - a. Scan dan data setiap input character dari awal hingga akhir file.
  - b. Hitung semua jumlah karakter dalam file input (n), dan data peluang tiap karakter yang muncul dalam file input (T).

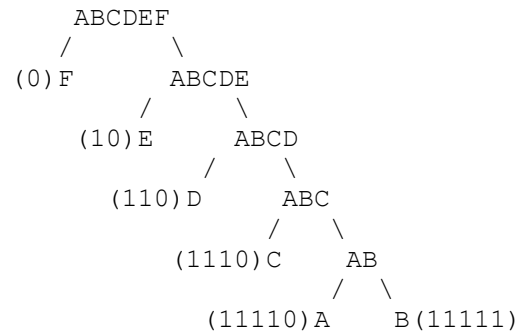
### 2. Konstruksi Pohon Huffman

Iterasi proses berikut sebanyak (n-1) kali :

- a. Urutkan karakter pada file mulai dari yang memiliki frekuensi paling kecil ke yang memiliki frekuensi paling besar.
  - b. Satukan dua karakter yang memiliki frekuensi paling kecil ( $m_1$  dan  $m_2$ ) menjadi pohon baru (P) dengan anggota yang lebih besar sebagai bagian kanan dari pohon.
  - c. Lakukan iterasi hingga semua anggota karakter tergabung.
3. Pada tahap ini, P hanya terdiri dari sebuah pohon yang beranggotakan seluruh karakter pada file tersebut. Berikan nilai bit 0 untuk anak sebelah kiri, dan bit 1 untuk anak sebelah kanan. Panjang kode untuk suatu simbol di Huffman adalah sebanyak berapa kali simbol tersebut bergabung dengan P di proses 2.

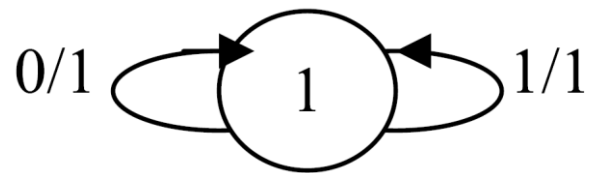
Berikut adalah contoh penerapan pohon Huffman dengan anggota sebagai berikut :

- |                  |                   |
|------------------|-------------------|
| A = 1 Kemunculan | D = 8 Kemunculan  |
| B = 2 Kemunculan | E = 16 Kemunculan |
| C = 4 Kemunculan | F = 32 Kemunculan |



## 2.2. Algoritma DMC

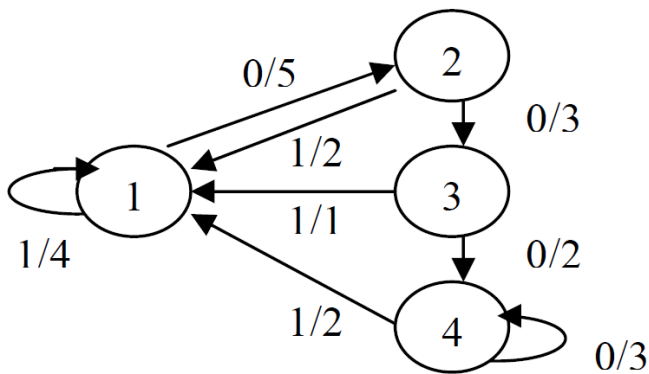
Algoritma DMC merupakan algoritma kompresi data yang memiliki jenis yang sama dengan algoritma Huffman yakni algoritma ini tidak menghilangkan memori objek yang akan dikompresi. Pada prosesnya algoritma DMC melakukan kompresi dengan penggunaan mesin FSA dengan adanya *inisial state* dan *final state*.



Gambar 1. State Pada Algoritma DMC dengan nilai bit 1

Seperti yang telah dijelaskan di atas, algoritma DMC menggunakan permodelan *finite state*. Gambar di atas menunjukkan FSA sebagai model awal DMC dengan nilai transisi 0/1 dan 1/1. Algoritma ini membuat probabilitas dari *precessor* dan *successor* karakter biner dalam file menggunakan permodelan *finite state*. Proses algoritma ini dikatakan adaptif/dinamik karena struktur mesin *finite state* akan berubah seiring dengan proses kompresi yang dilakukan.

Pada algoritma DMC ini, karakter dalam file diproses per bit. Perhitungan output yang dihasilkan dari setiap state akan digunakan untuk memperkirakan probabilitas dari transisi antar state. Sebagai contoh gambar di bawah, pada state 1, output diberi label 0/5. Label ini menjelaskan bit 0 pada state 1 terjadi sebanyak lima kali.



**Gambar 2. Ilustrasi Transisi antar state pada algoritma DMC**

Pada algoritma ini, transisi ditandai dengan  $0/p$  atau  $1/q$  dengan  $p$  dan  $q$  masing-masing menunjukkan jumlah transisi dari state dengan input 0 atau 1. Dari nilai  $p$  dan  $q$ , didapatkan nilai probabilitas untuk input selanjutnya, yaitu untuk input bernilai 0 adalah  $p/(p+q)$  dan untuk input bernilai 1 adalah  $q/(p+q)$ . Bila bit sesudah state tertentu bernilai 0, maka jumlah bit 0 pada transisi tersebut bertambah menjadi  $p+1$ , begitu pula bila bit sesudah state bernilai 1, jumlah bit 1 bertambah menjadi  $q+1$ .

Bila terjadi masalah suatu bit tidak muncul pada state tertentu, nilai bit pada state tersebut dapat ditangani dengan menginisialisasi state dengan nilai satu. Kemudian nilai probabilitas dihitung dengan menggunakan frekuensi relatif dari dua transisi output dari state yang baru.

Berikut source code untuk algoritma DMC :

```

1. s □ 1 /* jumlah state sekarang */
2. t □ 1 /* state sekarang */
3. T[1][0] = T[1][1] □ 1 /* model inisialisasi */
4. C[1][0] = C[1][1] □ 1 /* inisialisasi untuk menghindari masalah
frekuensi nol */
5. Untuk setiap input bit e :
   i. u □ t
   ii. t □ T[u][e] /*ikuti transisi*/
   iii. Kodekan e dengan probabilitas : C[u][e] / (C[u][0] +
C[u][1])
   iv. C[u][e] □ C[u][e]+1
   v. Jika ambang batas cloning tercapai, maka :
       □ s □ s + 1 /* state baru t' */
       □ T[u][e] □ s ; T[s][0] □ T[t][0] ; T[s][1] □
T[t][1]
       □ Pindahkan beberapa dari C[t] ke C[s]

```

### 3. PEMBAHASAN

Untuk mencari rute terpendek, dapat menggunakan beberapa metode, antara lain dengan menggunakan

algoritma Dijkstra, algoritma Bellman-ford, dan algoritma Warshall. Pada masing-masing sub-bab akan dijelaskan metode dan pseudocode dari masing-masing algoritma.

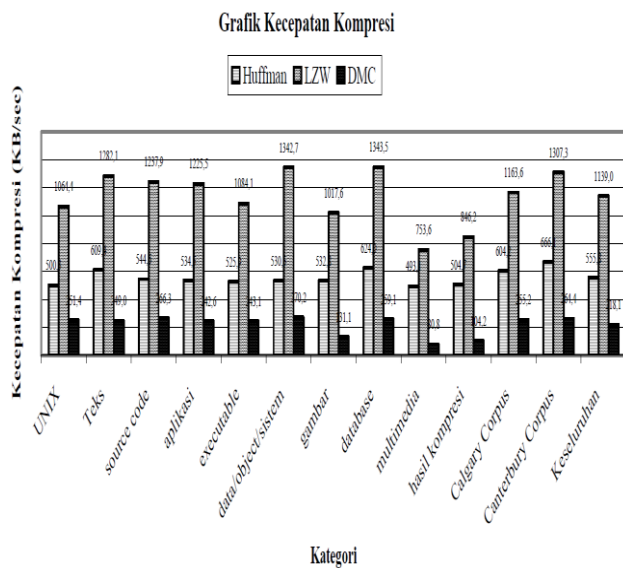
Algoritma Huffman dan DMC adalah dua algoritma kompresi data yang sudah terkenal dan banyak digunakan. Masing-masing dari algoritma tersebut memiliki karakteristik yang cukup berbeda dalam proses kompresi data. Kedua algoritma yang dibahas dalam makalah ini memiliki keuntungan tersendiri yakni kedua algoritma tersebut adalah algoritma kompresi data jenis *lossless compression scheme* dimana objek yang dikompresi tidak mengalami pengurangan / kehilangan data. Hal ini sangat menguntungkan terutama penggunaannya dalam peringkasan data untuk disimpan sehingga kedua algoritma tersebut banyak digunakan untuk meringkas data seperti pada aplikasi zip, rar, 7zip, dsb. Namun kedua algoritma tersebut tidak banyak ditemui pada *converter* file multimedia yang banyak menggunakan *lossy compression scheme*.

Berikut perbandingan kedua algoritma tersebut ditinjau dari kecepatan dan efektifitas pada proses kompresi data.

### 3.1 Perbandingan Efisiensi Waktu

Dari hasil eksplorasi di internet, penulis mendapatkan grafik perbandingan algoritma kompresi antara algoritma huffman, LZW, dan DMC. Grafik tersebut penulis lampirkan di bawah ini. Dalam grafik tersebut dicantumkan perbandingan kecepatan kompresi yang dinyatakan dalam satuan KB/s. Dari gambar tersebut, dapat dilihat bahwa Pengompresian data menggunakan metode huffman lebih cepat dibandingkan dengan menggunakan algoritma DMC. Hal tersebut ditunjukkan dengan grafik kecepatan yang lebih tinggi di semua aspek pengujian dari sumber yang penulis temui.

Bila kita melihat dari efektifitas masing-masing algoritma, hal ini tampak searah dan secara sekilas memberikan penjelasan yang logis. Pada algoritma Huffman, pemrosesan data bersifat statik dan menggunakan fungsi rekursif untuk realisasinya. Sedangkan bila kita melihat algoritma DMC yang bersifat dinamik, kita akan mendapatkan banyak sekali state yang harus dienkrpsi dan memungkinkan untuk terjadi proses 'bolak-balik' antar state dalam waktu cukup lama pada kasus tertentu.

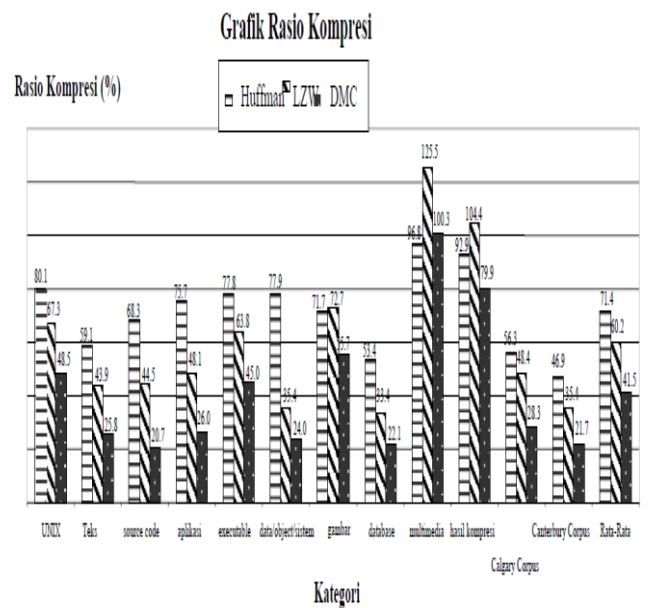


Gambar 3. Hasil Perbandingan kecepatan kompresi menggunakan algoritma huffman, LZW dan DMC

### 3.2 Perbandingan Efisiensi Kompresi

Berbeda dari hasil pada perbandingan efisiensi waktu pemrosesan kedua data tersebut, kompresi yang dihasilkan oleh kedua algoritma berbanding terbalik dengan yang terjadi pada perbandingan yang ditinjau dari waktu pemrosesan.

Dari grafik yang penulis dapatkan di internet, hasil kompresi dari kedua algoritma tersebut dimenangkan oleh algoritma DMC. Rasio perbandingan dalam persen yang dihasilkan oleh algoritma DMC lebih unggul di semua aspek dibandingkan dengan algoritma huffman. Pada algoritma huffman, kita melakukan enkripsi dengan pendataan seluruh karakter yang ada pada file sebelum melakukan proses kompresi. Sedangkan pada algoritma DMC, kita melihat bahwa input yang dilakukan cukup satu kali, kemudian untuk kalkulasi pengkodean dilakukan sambil berjalan perlahan. Hal ini memberikan efektifitas yang lebih tinggi dibandingkan dengan algoritma huffman. Bila kita sedikit menelusuri proses huffman, bit yang dihasilkan pada proses tersebut akan bertambah banyak seiring dengan karakter yang ada dalam file. Berbeda dengan proses Huffman yang melakukan proses pendataan satu per satu berdasarkan state yang dihasilkan dari input dan outputnya yang lebih hemat dalam penggunaan memori dan bit yang dihasilkan untuk masing-masing karakter.



Gambar 4. Hasil Rasio Konversi untuk algoritma huffman, LZW, dan DMC.

## IV. KESIMPULAN

Dalam dunia IT, terdapat banyak sekali algoritma yang bisa digunakan untuk melakukan kompresi data. Algoritma Huffman dan algoritma DMC hanyalah dua contoh dari sekian banyak algoritma yang tergolong ke dalam jenis *lossless compression scheme*. Algoritma ini tergolong cocok untuk melakukan proses peringkasan file. Namun kedua algoritma ini kurang cocok untuk melakukan proses konversi file ke dalam format lain dikarenakan kedua algoritma ini menjaga keutuhan file yang dikompresi. Selain itu, terdapat dua perbedaan yang cukup signifikan pada algoritma huffman dan algoritma DMC. Algoritma Huffman membutuhkan waktu untuk melakukan konversi jauh lebih cepat dibandingkan dengan algoritma DMC. Sedangkan algoritma DMC memiliki keunggulan dalam hasil kompresi yang lebih baik dengan rasio kompresi yang ditunjukkan lebih unggul dibandingkan dengan algoritma huffman di semua pengujian sumber. Kedua algoritma tersebut memiliki keuntungan masing-masing dalam aspek yang berbeda. Hal tersebut akan menjadi pertimbangan untuk menghasilkan algoritma yang tepat dalam membuat program untuk meringkas file supaya program dapat memberikan solusi atas masalah secara tepat dan efektif sesuai yang diinginkan.

## REFERENSI

- [1] <http://michael.dipperstein.com/huffman/> (15/12/2010 23:55).
- [2] [http://en.wikipedia.org/wiki/Huffman\\_coding](http://en.wikipedia.org/wiki/Huffman_coding) (15/12/2010 23:58)
- [3] <http://adjiedjokers.wordpress.com/2010/04/28/tentang-coding-huffman/> (16/12/2010 00:01)
- [4] <http://arraydalamprogram.blogspot.com/2010/03/huffman-coding.html> (16/12/2010 00:03)
- [5] [http://en.wikipedia.org/wiki/Data\\_compression#Lossy\\_data\\_compression](http://en.wikipedia.org/wiki/Data_compression#Lossy_data_compression) (16/12/2010 00:18)
- [6] [http://en.wikipedia.org/wiki/Dynamic\\_Markov\\_Compression](http://en.wikipedia.org/wiki/Dynamic_Markov_Compression)(16/12/2010 00:25)
- [7] <http://elib.unikom.ac.id/gdl.php?mod=browse&op=read&id=jbptunikompp-gdl-ikinim1010-14966> (16/12/2010 00:35)
- [8] <http://elib.unikom.ac.id/gdl.php?mod=browse&op=read&id=jbptunikompp-gdl-ikinim1010-14966> (16/12/2010 00:40)
- [9] <http://www.tcm.phy.cam.ac.uk/~ajw29/thesis/node28.html> (16/12/2010 00:45)

## 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 2010

ttd

Emil Fahmi Yakhya  
13509069