

## KODE HUFFMAN UNTUK KOMPRESI PESAN

Erna Zuni Astuti<sup>1</sup>, Erwin Yudi Hidayat<sup>2</sup>

<sup>1,2</sup>Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro  
Jalan Nakula I No. 5-11, Semarang, 50131, (024) 3517261  
E-mail : ernazunias@yahoo.com<sup>1</sup>, erwin@research.dinus.ac.id<sup>2</sup>

---

### **Abstrak**

*Dalam ilmu komunikasi data, pesan yang dikirim kepada seseorang, seringkali ukurannya terlalu besar, sehingga membutuhkan tempat penyimpanan yang terlalu besar pula. Demikian juga pesan yang terlalu besar, akan membutuhkan waktu pengiriman yang lebih lama bila dibandingkan dengan pesan yang berukuran relatif lebih kecil. Dua masalah tersebut di atas, sebenarnya bisa diatasi dengan pengkodean pesan dengan tujuan agar isi pesan yang sebenarnya besar, bisa dibuat sesingkat mungkin sehingga waktu pengiriman yang mestinya lama bisa dibuat relatif lebih cepat dan tempat penyimpanan yang besar bisa dibuat relatif lebih efisien dibandingkan dengan sebelum dilakukan pengkodean. Dari Hasil uji coba penerapan dan penghitungan kode Huffman, maka dapat disimpulkan antara lain bahwa dengan menggunakan kode Huffman ternyata dapat mengurangi beban alias dapat mengkompres data lebih dari 50%.*

**Kata Kunci:** Kode Huffman, Kompresi Pesan, Komunikasi

### **Abstract**

*In the field of data communication, messages that sent to receivers are often too large in size, causes larger size of storage are needed. As for that, extra time for sending is required. Those both problems could be solved by message coding using Huffman Code, to decrease the size of message. The purposes are to provide shorter time for message sending and to save storage space. Result shows that Huffman Coding was able to cut the loads or compress data more than 50%.*

**Keywords:** Huffman code, Message Compression, Communication

## **1. LATAR BELAKANG**

Dalam berkomunikasi, seseorang terkadang tidak pernah berpikir bagaimana terjadinya penyimpanan data, berapa besar tempat yang dibutuhkan untuk menyimpan data tersebut dan berapa lama waktu yang dibutuhkan dari mereka menulis sampai terkirimnya suatu pesan ketempat yang dituju. Orang awam tidak pernah mau mengerti apa, mengapa dan bagaimana ilmu tentang pengiriman data dan penyimpanan data. Bagi mereka yang penting adalah apa yang mereka inginkan dapat terpenuhi dan dapat

dicapai dalam waktu yang singkat dan hasilnya akurat dan juga memuaskan [1] Oleh karena itu kita sebagai ilmuwan kecil harus tanggap atas semua keadaan tersebut di atas, karena sebenarnya semua permasalahan di atas memang bukan permasalahan mereka. Tetapi permasalahan di atas adalah permasalahan kita, yang harus kita pecahkan sedemikian rupa, sesuai dengan ilmu pengetahuan yang kita miliki saat ini.

Dalam ilmu Komunikasi data, pesan yang dikirim kepada seseorang, seringkali ukurannya terlalu besar, sehingga membutuhkan tempat

penyimpanan yang terlalu besar pula. Demikian juga pesan yang terlalu besar, akan membutuhkan waktu pengiriman yang lebih lama bila dibandingkan dengan pesan yang berukuran relative lebih kecil.

Dua masalah tersebut di atas, sebenarnya bisa diatasi dengan pengkodean pesan dengan tujuan agar isi pesan yang sebenarnya besar, bisa dibuat sesingkat mungkin sehingga waktu pengiriman yang mestinya lama bisa dibuat relatif lebih cepat dan tempat penyimpanan yang besar bisa dibuat relatif lebih efisien dibandingkan dengan sebelum dilakukan pengkodean. Berdasarkan latarbelakang permasalahan di atas, maka penulis tertarik untuk melakukan pembahasan dan penulisan serta tata cara pemampatan data atau yang lebih terkenal dengan istilah kompresi data dengan mengambil judul Kode Huffman untuk pemampatan/Kompresi Data.

## 2. RUMUSAN MASALAH

Bagaimana cara menggunakan kode Huffman untuk pemampatan atau Compression data agar permasalahan seperti diatas bisa teratasi dengan baik

## 3. BATASAN MASALAH

- Pesan yang akan dibahas dalam penulisan ini terbatas pada pesan teks saja
- Huruf capital/huruf besar, huruf kecil maupun latin dianggap sama
- Tidak memandang karakter lain seperti titik, koma dan tanda baca lainnya
- Setiap pesan dianggap tidak terdapat spasi

## 4. TUJUAN

- Menerapkan kode Huffman untuk memampatkan/mengkompresi data, agar dapat tercapai:
  - Ruang penyimpanan yang relative lebih kecil dibandingkan sebelumnya
  - Waktu pengiriman pesan relative lebih cepat dibandingkan sebelumnya
- Pengkodean dilakukan dengan cara pengkodean setiap karakter di dalam pesan atau di dalam arsip dikodekan dengan kode yang lebih pendek.

## 5. LANDASAN TEORI

Sistem kode yang selama ini banyak diterapkan dan umum digunakan serta paling populer di dalam dunia ilmu pengetahuan saat ini adalah kode ASCII (American Standard Code International Institute). Cara kerja dari pengkodean ASCII ini adalah dengan cara mengkodekan setiap karakter menjadi 8 bit Biner, atau sering disebut dengan istilah satu karakter dikodekan dengan satu byte [1].

Untuk mengetahui gambaran dan cara kerja pengkodean ini, misalkan kita ambil beberapa huruf sebagai contoh adalah sebagai berikut:

Untuk kode ASCII huruf A adalah = 01000001, untuk huruf B=01000010, untuk huruf C=01000011 sedangkan untuk huruf D adalah = 01000100 dst.

**Tabel 1:** Contoh Kode ASCII dari huruf

No	Huruf	Kode ASCII
1	A	01000001
2	B	01000010
3	C	01000011
4	D	01000100

Apabila kita mau menuliskan kata atau pesan seperti berikut ini, misalnya: ABACBCDACB, maka untuk pertama kalinya kita harus melihat berapa banyaknya karakter setiap pesan yang mau dikodekan. Andaikan pesan diatas yang akan dikodekan, maka kita lihat pesan tersebut terdiri dari 10 karakter, maka secara otomatis akan dikodekan dalam ASCII menjadi  $8 \times 10 = 80$  bit (10 byte). Kira-kira seperti ini hasil pengkodeannya:

ABACBCDACB = 01000001 01000010  
01000001 01000011 01000010  
0100001101000100  
010000010100001101000010.

Untuk 10 karakter saja pengkodean ASCII sudah sebegitu panjangnya. Apalagi untuk pengkodean suatu kalimat atau untuk suatu pesan yang sangat panjang, Oleh karena itu perlu diusahakan untuk meminimumkan jumlah bit yang dibutuhkan, panjangnya kode untuk setiap karakter sedapat mungkin diperpendek, agar dibutuhkan tempat penyimpanan yang relative lebih sederhana, efektif dan efisien. Demikian juga untuk waktu pengiriman yang dibutuhkan.

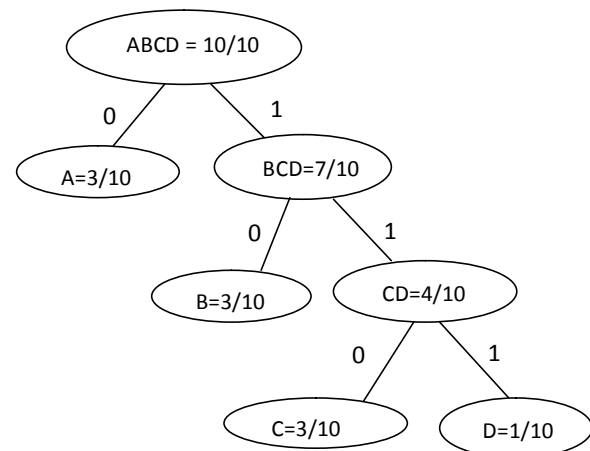
Pengkodean ini terutama ditujukan untuk karakter-karakter yang probabilitas/peluang kemunculannya lebih besar. maka harus diusahakan agar pengkodeannya lebih pendek jika dibandingkan dengan karakter yang probabilitas kemunculannya lebih kecil [1]. Dasar pemikiran seperti inilah yang mendasari munculnya ide baru, yaitu membuat kode yang lain selain kode ASCII, yaitu dengan menggunakan kode Huffman. Sehingga kita akan mengusahakan untuk meminimalkan pengkodean tersebut menggunakan kode Huffman.

### 5.1 Langkah Kerja Huffman

Langkah-langkah yang harus dilakukan sebelum mengerjakan pengkodean Huffman adalah sebagai berikut:

1. Tentukan jumlah semua karakter yang ada dalam suatu pesan yang akan dikodekan.
2. Tentukan kerapatannya, yaitu berapa kali huruf tersebut muncul di dalam kalimat/string yang akan di kodekan
3. Tentukan probabilitas/peluang setiap karakter, yaitu banyaknya kerapatan dibagi jumlah seluruh karakter
4. Tentukan kode Huffman untuk setiap karakternya, dengan cara membuat pohon biner
5. Karakter yang kerapatannya lebih besar, harus diusahakan dengan kode yang lebih pendek
6. Buatlah table untuk meringkas sekaligus melihat hasilnya

Contoh pohon biner di bawah ini adalah contoh pembuatan pohon biner untuk mendapatkan kode Huffman suatu pesan ABACBCDACB adalah sebagai berikut [4]:



Gambar 1. Pohon Huffman untuk pesan ABACBCDACB

**Tabel 2:** Tabel Kerapatan kode Huffman ABACBCDACB [2]

Simbol	Kerapatan	Probabilitas	Kode Huffman
A	3	3/10	0
B	3	3/10	10
C	3	3/10	110
D	1	1/10	111

Simbol-simbol yang sering muncul di dalam kalimat/string direpresentasikan dengan kode yang lebih pendek, dari pada kode untuk symbol yang jarang muncul seperti tampak pada gambar dan tabel di atas. Kode Huffman tidak bersifat unik, artinya kode untuk setiap karakter berbeda-beda pada setiap pesan tergantung pada kerapatan kemunculan karakter pada setiap pesan. Demikian juga dengan keputusan apakah akan diletakkan di sebelah kanan atau disebelah kiri akan memberikan kode yang berbeda-beda, tetapi tetap saja tidak akan pernah mengubah panjang kode yang diperolehnya secara keseluruhan [1].

Dengan menggunakan kode Huffman dalam table di atas, maka pesan yang bertuliskan ABACBCDACB dapat dinyatakan dalam rangkaian bit menjadi sbb:

ABACBCDACB=

010011010110111011010 = 21 bit saja.

Bila dibandingkan dengan kode ASCII yang sudah di buat di atas, maka bisa dikompres dari 80 bit menjadi 21 bit saja. Jika dihitung secara prosentase penyusutan adalah  $21/80 * 100\% = 26,25\%$ .

Jadi dengan menggunakan kode Huffman bisa terlihat dengan jelas bahwa jumlah bit yang dibutuhkan benar-benar sudah berkurang secara signifikan. Untuk contoh di atas bias dikatakan penyusutannya lebih dari 70%.

## 6. IMPLEMENTASI

### 6.1 Contoh Penerapan

Agar bisa menerapkan kode Huffmann, kita akan menguji coba dasar teori yang sudah dijelaskan di atas untuk diterapkan dalam suatu kalimat. Misalnya kalimatnya seperti di bawah ini:

Gunakan kode Huffmann untuk mengkompres suatu pesan yang dikirimkan kepada seseorang dengan susunan kalimat sebagai berikut: “HARI INI KITA MAKAN SIANG DI MANA” [4]

Untuk mendapatkan kode Huffmann kita harus mengikuti langkah-langkah sesuai dengan petunjuk:

1. Menentukan Probabilitas atau peluang dari kemunculan setiap karakter
2. Hasil penghitungannya diletakkan dalam table
3. Membuat pohon biner
4. Menentukan kode setiap karakter
5. Menghitung berapa banyaknya bit yang digunakan
6. Membandingkan hasil Kode Huffmann dengan kode ASCII

Selanjutnya kita akan melakukan penghitungan satu per satu untuk setiap karakter:

#### Langkah Pertama:

Menentukan kerapatan masing-masing karakter dalam susunan kalimat di atas Banyaknya huruf keseluruhan adalah = 27 karakter terdiri dari:

Huruf “A” = 7, Huruf “I” = 6, Huruf “N”= 4, Huruf “M” = 2 , Huruf “K” = 2 , Huruf “H” = 1, Huruf “ R” = 1, Huruf “T” = 1, Huruf “ S” = 1, Huruf “G” = 1 dan Huruf “ D” = 1 Sehingga

**Langkah kedua:**

Menentukan probabilitas Masing-masing karakter adalah sebagai berikut [2]:

Huruf "A" =  $7/27$ , Huruf "I" =  $6/27$ ,  
Huruf "N" =  $4/27$ , Huruf "M" =  $2/27$ ,  
Huruf "K" =  $2/27$ , Huruf "H" =  $1/27$ ,  
Huruf "R" =  $1/27$ , Huruf "T" =  $1/27$ ,  
Huruf "S" =  $1/27$ , Huruf "G" =  $1/27$ ,  
Huruf "D" =  $1/27$ .

Apabila hasil tersebut ditabulasikan menjadi seperti pada tabel 3.

**Langkah ketiga:**

Membuat tabel seperti Berikut:

**Tabel 3:** Tabel Kerapatan dan Peluang untuk setiap karakter [2]

Huruf	Kerapatan	Peluang
A	7	$7/27$
I	6	$6/27$
N	4	$4/27$
M	2	$2/27$
K	2	$2/27$
H	1	$1/27$
R	1	$1/27$
T	1	$1/27$
S	1	$1/27$
G	1	$1/27$
D	1	$1/27$

**Langkah ke empat:**

Membuat pohon biner seperti pada Gambar 2 [3].

**Langkah kelima:**

Ringkasan hasil pembuatan pohon biner untuk setiap karakter adalah sebagai berikut [3]:

A = 00                      M = 110  
I = 010                     K = 1110  
S = 0110                  H = 11110  
G = 01110                R = 111110  
D = 01111                T = 111111  
N = 10

Bila dibuat tabulasi bisa dinyatakan sebagai berikut:

**Tabel 4:** Kode Huffman untuk Kerapatan dan Peluang setiap karakter

Huruf	Kerapatan	Peluang	Kode Huffman
A	7	$7/27$	00
I	6	$6/27$	010
M	4	$4/27$	110
N	2	$2/27$	10
K	2	$2/27$	1110
H	1	$1/27$	11110
R	1	$1/27$	111110
T	1	$1/27$	111111
S	1	$1/27$	0110
G	1	$1/27$	01110
D	1	$1/27$	01111

**Langkah keenam:**

Hasil rekapitulasi akhir adalah sebagai berikut [4]:

a. Banyaknya bit untuk Huruf "A"  
" =  $2 * 7 = 14$

b. Banyaknya bit untuk Huruf "I"  
" =  $3 * 6 = 18$

c. Banyaknya bit untuk Huruf "N"  
" =  $2 * 4 = 8$

d. Banyaknya bit untuk Huruf "M"  
" =  $3 * 2 = 6$

e. Banyaknya bit untuk Huruf "K"  
" =  $4 * 2 = 8$

f. Banyaknya bit untuk Huruf "H"  
" =  $5 * 1 = 5$

g. Banyaknya bit untuk Huruf "R"  
" =  $6 * 1 = 6$

h. Banyaknya bit untuk Huruf "T"  
" =  $6 * 1 = 6$

i. Banyaknya bit untuk Huruf "S"  
" =  $4 * 1 = 4$

j. Banyaknya bit untuk Huruf "G"  
" =  $5 * 1 = 5$

k. Banyaknya bit untuk Huruf "D"  
" =  $5 * 1 = 5$

**Hasil akhir secara keseluruhan:**

Kalimat yang di kodekan adalah:  
 “HARI INI KITA MAKAN SIANG DI  
 MANA” [5]

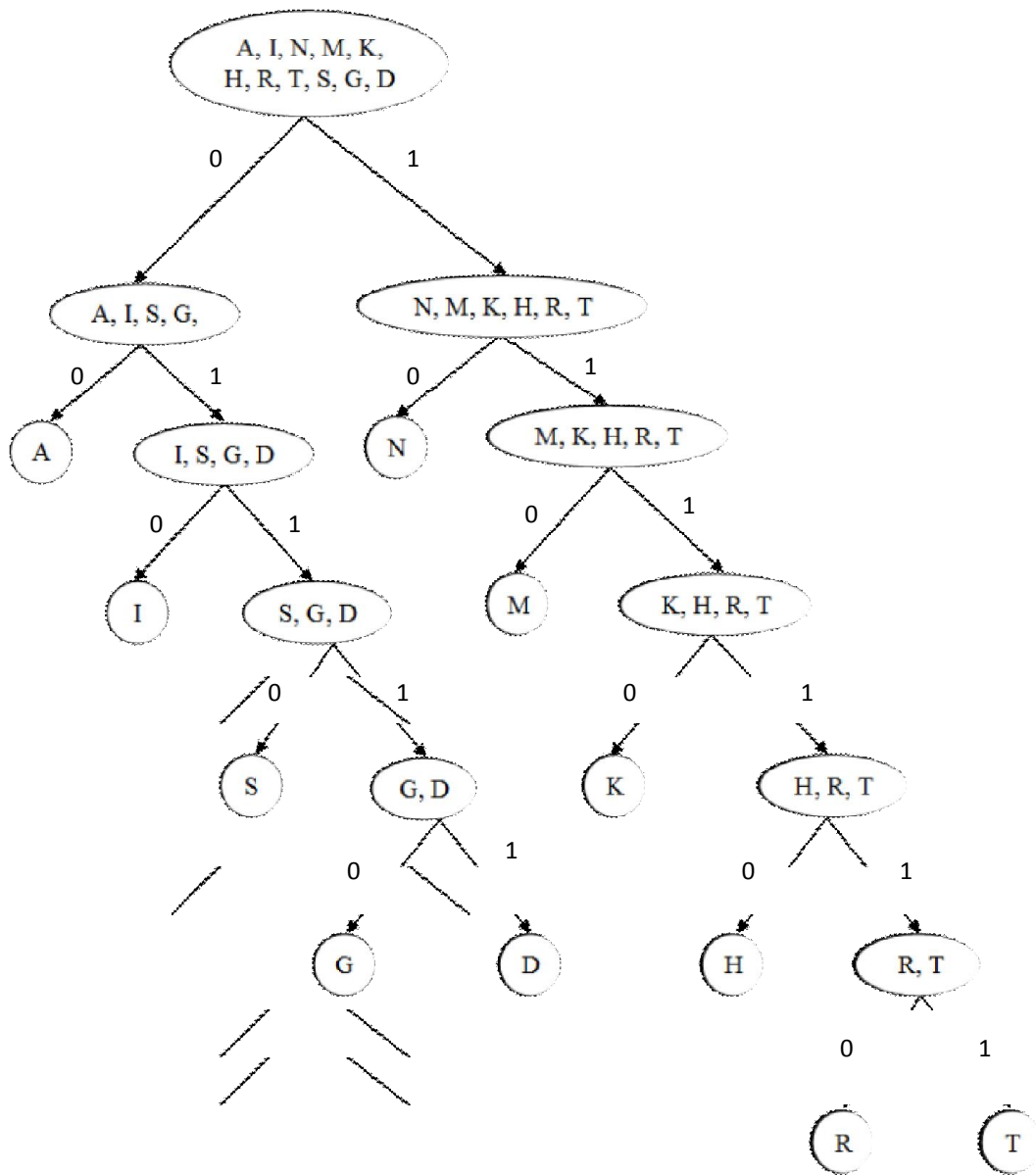
Dengan Kode ASCII:  $27 * 8 \text{ Bit} = 216 \text{ Bit}$ .

Dengan Kode Huffmann 85 Bit adalah  
 $85/216 * 100\% = 39,35 \%$ .

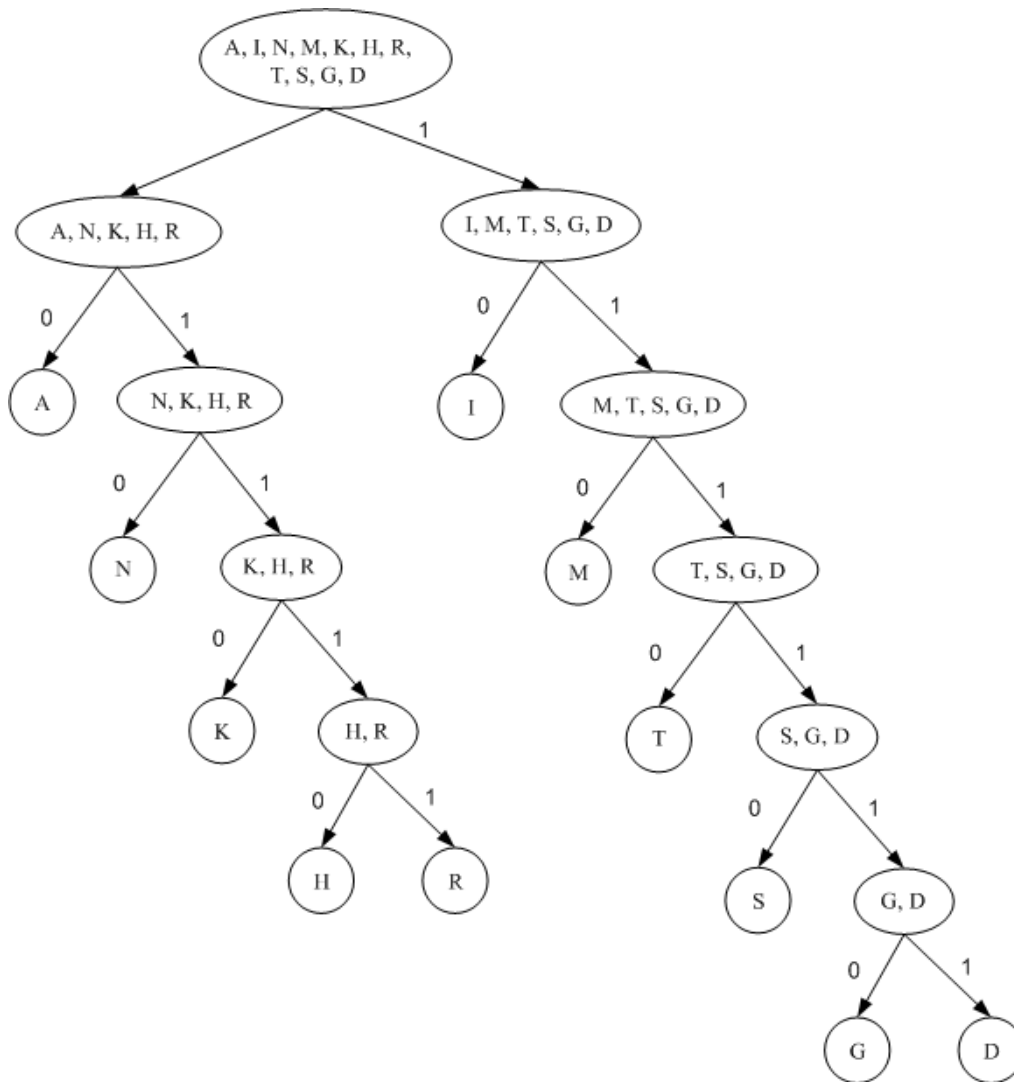
Sehingga dengan kode Huffmann bisa  
 mengurangi beban lebih dari 60 %.

**6.2 Kode Huffman Tidak Unik**

Untuk membuktikan bahwa Kode Huffman tidak unik, bisa dilakukan dengan membuat pohon biner seperti pada Gambar 3.



**Gambar 2.** Pohon biner untuk kalimat  
 “HARI INI KITA MAKAN SIANG DI MANA”



**Gambar 3.** Pohon biner untuk membuktikan kode Huffman tidak unik

Ringkasan hasil pembuatan pohon Biner untuk setiap karakter pada gambar 3 adalah sebagai berikut:

- A = 00
- N = 010
- K = 0110
- H = 01110
- R = 01111
- I = 10
- M = 110
- T = 1110
- S = 11110
- G = 111110
- D = 111111

Dari pohon biner pada gambar 3 di atas, bila dibuat tabulasi bisa dinyatakan seperti pada tabel 5 berikut:

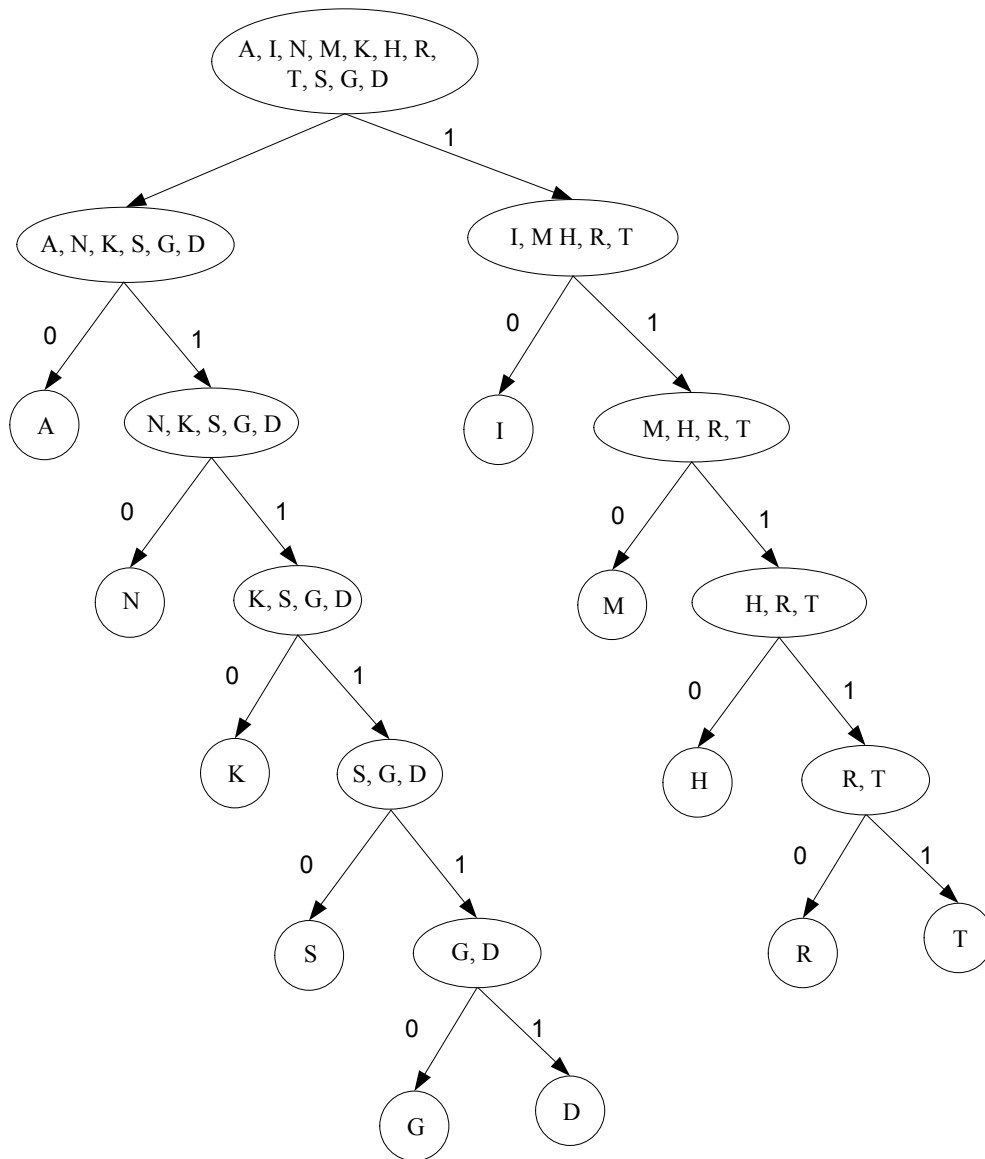
**Tabel 5:** Kode Huffman untuk Kerapatan dan Peluang setiap karakter dari gambar 3

Huruf	Kerapatan	Peluang	Kode Huffman
A	7	7/27	00
I	6	6/27	10
M	4	4/27	110
N	2	2/27	010
K	2	2/27	0110
H	1	1/27	01110
R	1	1/27	01111

T	1	1/27	1110
S	1	1/27	11110
G	1	1/27	111110
D	1	1/27	111111

Hasil rekapitulasi akhir adalah sebagai berikut [4]:

- a. Banyaknya bit untuk Huruf “ A “ =  $2 * 7 = 14$
- b. Banyaknya bit untuk Huruf “ I “ =  $2 * 6 = 12$



**Gambar 4.** Contoh lain untuk membuktikan kode Huffman tidak unik



- c. Banyaknya bit untuk Huruf “ N  
“ =  $3 * 2 = 6$
- d. Banyaknya bit untuk Huruf “ M  
“ =  $3 * 4 = 12$
- e. Banyaknya bit untuk Huruf “ K  
“ =  $2 * 4 = 8$
- f. Banyaknya bit untuk Huruf “ H  
“ =  $1 * 5 = 5$
- g. Banyaknya bit untuk Huruf “ R  
“ =  $1 * 5 = 5$
- h. Banyaknya bit untuk Huruf “ T  
“ =  $1 * 4 = 4$
- i. Banyaknya bit untuk Huruf “ S  
“ =  $1 * 5 = 5$
- j. Banyaknya bit untuk Huruf “ G  
“ =  $1 * 6 = 6$
- k. Banyaknya bit untuk Huruf “ D  
“ =  $1 * 6 = 6$

Maka, perbandingan hasil akhir secara keseluruhan adalah dengan kode ASCII:  $27 * 8 \text{ Bit} = 216 \text{ Bit}$ . Dengan Kode Huffmann 83 Bit adalah  $83/216 * 100\% = 38,43 \%$ . Dengan demikian, kode Huffman terbukti mengurangi beban secara signifikan.

Contoh lain untuk membuktikan kode Huffman tidak unik diperlihatkan pada gambar 4.

Ringkasan hasil pembuatan pohon Biner untuk setiap karakter pada gambar 4 adalah sebagai berikut:

A = 00                      I = 10  
 N = 010                    M = 110  
 K = 0110                 H = 1110  
 S = 01110                R = 11110  
 G = 011110              T = 11111  
 D = 011111

Tabel 6 berikut menunjukkan representasi kerapatan dan peluang kode Huffman dari gambar 4.

**Tabel 6:** Kode Huffman untuk Kerapatan dan Peluang setiap karakter dari gambar 4

Huruf	Kerapatan	Peluang	Kode Huffman
A	7	7/27	00
I	6	6/27	10
M	4	4/27	110
N	2	2/27	010
K	2	2/27	0110
H	1	1/27	1110
R	1	1/27	11110
T	1	1/27	11111
S	1	1/27	01110
G	1	1/27	011110
D	1	1/27	011111

Hasil Rekapitulasi akhir adalah sebagai berikut:

- a. Banyaknya bit untuk Huruf “ A  
“ =  $2 * 7 = 14$
- b. Banyaknya bit untuk Huruf “ I  
“ =  $2 * 6 = 12$
- c. Banyaknya bit untuk Huruf “ N  
“ =  $3 * 2 = 6$
- d. Banyaknya bit untuk Huruf “ M  
“ =  $3 * 4 = 12$
- e. Banyaknya bit untuk Huruf “ K  
“ =  $4 * 2 = 8$
- f. Banyaknya bit untuk Huruf “ H  
“ =  $4 * 1 = 4$
- g. Banyaknya bit untuk Huruf “ R  
“ =  $5 * 1 = 5$
- h. Banyaknya bit untuk Huruf “ T  
“ =  $5 * 1 = 5$
- i. Banyaknya bit untuk Huruf “ S  
“ =  $5 * 1 = 5$
- j. Banyaknya bit untuk Huruf “ G  
“ =  $6 * 1 = 6$
- k. Banyaknya bit untuk Huruf “ D  
“ =  $6 * 1 = 6$

Perhitungan hasilnya adalah sebagai berikut:

Dengan kode ASCII:  $27 * 8 \text{ Bit} = 216 \text{ Bit}$ , dengan Kode Huffmann 93 Bit adalah  $93/216 * 100\% = 43,056 \%$ .

## 7. KESIMPULAN DAN SARAN

### 7.1 Kesimpulan

Dari Hasil uji coba penerapan dan Penghitungan kode Huffman, maka dapat disimpulkan antara lain bahwa:

1. Kode Huffmann benar-benar lebih sederhana bila di bandingkan dengan kode ASCII yang selama ini dipakai masyarakat umum.
2. Dengan menggunakan kode Huffmann ternyata dapat mengurangi beban alias dapat mengkompres data lebih dari 50%.
3. Penghitungan kode Huffmann tidak unik, tetapi bias bermacam –macam variasi dalam setiap kalimat yang dikodekan, tetapi semua hasil yang dicapai tetap lebih rendah dari pada kode ASCII.

### 7.2 Saran

Adapun saran yang bisa dicatat oleh penulis antara lain adalah sebagai berikut:

1. Karena kode Huffmann tidak unik, maka hasilnya bias berbeda beda antara orang yang satu dengan yang lainnya, sehingga meskipun lebih sederhana tetapi pada kenyataannya lebih susah untuk diterapkan.
2. Proses dan langkah-langkah untuk mendapatkan kode Huffmann lebih panjangdan rumit sehingga tidak disukai oleh para pengguna.

Priority Queues Wikipedia the free encyclopedia Maret 2015

- [4] Huffman Code Discussion and Implementation by Michael Dipperstein Wikipedia Maret 2013.

## DAFTAR PUSTAKA

- [1] Rinaldi Munir: Buku Teks Ilmu Komputer “Matematika Diskrit” Edisi Ketiga Tahun 2005 Penerbit Informatika Bandung
- [2] Sudjana “Metoda Statistika” Edisi Ke Enam Tahun 2005 Penerbit Tarsito Bandung
- [3] Huffman Coding : An Application of Binary Trees and