

## APLIKASI GRAF POHON PADA ALGORITMA HUFFMAN

Amrullah

Program Studi Pendidikan Matematika, Jurusan PMIPA FKIP Universitas Mataram  
Jl. Majapahit 62 Mataram, 83125  
amrullah.mtk@gmail.com

---

**Abstrak :** Sistem bilangan biner banyak digunakan dalam teknologi komputer. Bilangan biner digunakan oleh algoritma Huffman pada graf pohon. Algoritma Huffman menggunakan prinsip pengkodean yang mirip dengan kode Morse, yaitu tiap karakter dikodekan hanya dengan rangkaian beberapa bit 0 dan 1 sesuai distribusi kemunculan karakter dalam naskah. Sedang graf pohon adalah graf tak-berarah terhubung yang tidak memuat sirkuit sederhana. Setiap karakter dalam komputer memiliki panjang 8 bit. Dalam algoritma Huffman, dengan penggunaan graf pohon dari distribusi frekuensi karakter dalam suatu pesan, ukuran panjang bit setiap karakter dapat memiliki ukuran lebih kecil dari 8 bit sehingga dapat menghemat ukuran data. Pada akhirnya dapat disimpulkan bahwa algoritma Huffman dapat digunakan untuk menghemat ukuran data dalam sistem komputer.

**Kata Kunci :** Huffman, graf pohon, algoritma, biner

---

**Abstract :** Binary number system is more used in computer technology. Binary number used by the Huffman algorithm on the tree in graph theory. Huffman algorithm using the principle similar to Morse code, ie each character is encoded only by a series of multiple bits 0 and 1 corresponds to the distribution of the appearance of characters in the script. but tree is non connected graph that contains no simple circuit. Each character in the computer has a length of 8 bits. In the Huffman algorithm, with the use of a tree graph of the frequency distribution of characters in data, the size of the bit length of each character can have a size smaller than 8 bits so that it can save data size. In the finally it can be concluded that the Huffman algorithm can be used to save data size in the computer system.

**Keywords :** Huffman, tree, Algorithm, biner

---

### 1. PENDAHULUAN

Jaringan komputer mulai berkembang di awal tahun 1980 sebagai media komunikasi komputer yang berkembang pesat. Sampai saat ini komputer menjadi sarana komunikasi yang sangat efektif dan hampir seluruh bentuk informasi melibatkan komputer dalam penggunaannya. Setelah ditemukannya internet, berbagai informasi bisa diakses dari rumah dengan biaya yang murah. Komunikasi data sebenarnya merupakan gabungan dua teknik yang sama sekali jauh berbeda yaitu pengolahan data dan telekomunikasi, dapat diartikan bahwa komunikasi data memberikan layanan komunikasi jarak jauh dengan sistem komputer.

Pengiriman data antar komputer sudah menjadi kebutuhan utama dalam penggunaan komputer saat ini, sehingga berbagai teknologi dikembangkan untuk meningkatkan kecepatan pengiriman data, baik dengan teknologi hardware maupun teknologi software. Salah satu teknologi software yang berkembang saat ini adalah teknologi pemampatan data atau lebih dikenal dengan istilah kompres data. Berbagai cara dan algoritma pun telah dikembangkan untuk meningkatkan penghematan pengiriman data, antara lain algoritma Huffman. Algoritma ini dikenal dalam konsep teori graf.

Graf merupakan salah satu teori kombinatorik yang dapat diaplikasi pada jaringan komputer. Penemu graf adalah Leonhard Euler. Pada tahun 1736, graf ditemukan disebuah jembatan Königsberg. Di kota Königsberg, sebelah timur negara bagian Prussia Jerman, yang sekarang bernama kota Kaliningrad, terdapat sungai Pregal yg mengalir mengitari pulau Kneiphof lalu bercabang menjadi dua buah anak sungai. Ada 7 buah

jembatan yg menghubungkan daratan yg dibelah oleh sungai tersebut. Pada saat itu muncul suatu keinginan untuk dapat menikmati daerah tersebut dengan melalui ketujuh jembatan tepat satu kali, yakni bermula dari satu tempat dan kembali ke tempat semula. Mereka berusaha untuk memperoleh rute yang sesuai dengan keinginan tersebut, dengan selalu mencoba menjalaninya. Masalah ini, dikenal dengan nama masalah Euler yang diselesaikan dengan konsep Euler dalam teori graf [1].

Graf dapat diaplikasi pada berbagai bidang seperti penemuan jarak terpendek, pelabelan, pewarnaan, maupun pemampatan data. Pada pemampatan data, dikenal algoritma Huffman. Algoritma Huffman merupakan algoritma yang memperkecil ukuran bit suatu karakter berdasarkan distribusi kemunculan karakter dalam naskah. Pengkodean Huffman ini didasarkan pada kekerapan kemunculan suatu simbol pada data sumber yang akan dimampatkan. Hasil dari pengkodean ini akan menghasilkan kode awalan (prefix code) untuk setiap simbol dimana simbol yang kekerapan kemunculannya lebih besar memiliki kode awalan yang lebih pendek dari simbol yang memiliki kekerapan kemunculannya lebih kecil pada data sumber. Dalam pembangkitan kode Huffman, setiap karakter disusun dalam bentuk graf pohon.

Adapun tujuan tulisan ini adalah untuk menerapkan Algoritma Huffman dapat memperkecil ukuran bit setiap karakter dalam sistem komputer, menerapkan graf pohon dalam algoritma Huffman untuk memampatkan data.

### II. PEMBAHASAN

Untuk menerapkan graf pohon dalam algoritma Huffman, dilakukan prosedur sebagai berikut (1) Kajian pustakan tentang graf pohon, disini akan dibahas tentang definisi

dan sifat dari graf pohon, demikian juga tentang spanning tree. (2) Kajian pustakan tentang Kode Biner, disini akan dibahas definisi kode biner, system bilangan biner dihubungkan dengan sistem komputer (3) Algoritma Huffman, disini dibahas tentang langkah algoritma Huffman, bagaimana membangun graf pohon dari algoritma Huffman. Disini juga dibahas bagaimana pesan dibentuk menjadi graf pohon sehingga membentuk pesan dengan jumlah bit yang kecil.

**Graf Pohon**

Pohon (tree) adalah graf tak-berarah terhubung yang tidak memuat sirkuit sederhana[2].

Menurut definisi di atas, suatu pohon merupakan graf yang terhubung, tak berarah, dan tidak memuat sirkuit, ini merupakan tiga kriteria utama untuk pohon. Pohon adalah bentuk khusus dari suatu graf yang banyak diterapkan untuk berbagai keperluan. Misalnya struktur organisasi suatu perusahaan, silsilah suatu keluarga, skema sistem gugur suatu pertandingan, dan ikatan kimia suatu molekul adalah jenis graf yang tergolong sebagai pohon. Pada pohon, simpul-simpul yang berderajat satu dinamakan daun (leave), sedangkan simpul yang derajatnya lebih besar daripada satu dinamakan simpul cabang (branch node) atau simpul internal (internal node) dan kumpulan pohon-pohon yang terpisahkan satu sama lain disebut hutan (forest) [2].

Dalam struktur data, pohon memegang peranan yang cukup penting. Struktur ini biasanya digunakan terutama untuk menyajikan data yang mengandung hubungan hierarchyal antara elemen-elemen mereka.

Bentuk pohon khusus yang lebih mudah dikelola dalam komputer adalah pohon binary. Bentuk ini merupakan bentuk pohon yang umum. Sebuah pohon binar T didefinisikan terdiri dari sebuah himpunan hingga elemen yang disebut simpul [2], sedemikian sehingga :

- a. T adalah hampa (disebut pohon null) atau
- b. T mengandung simpul R yang dipilih disebut akar (root) dari T, dan simpul sisanya membentuk 2 pohon binary T1 dan T2 yang saling lepas.

**Sifat-sifat Pohon**

Misalkan  $G = (V, E)$  adalah graf tak-berarah sederhana dan jumlah simpulnya  $n$ . Maka, semua pernyataan di bawah ini adalah ekuivalen [3]:

- 1.  $G$  adalah pohon.
- 2. Setiap pasang simpul di dalam  $G$  terhubung dengan lintasan tunggal.
- 3.  $G$  terhubung dan memiliki  $m = n - 1$  buah sisi.
- 4.  $G$  tidak mengandung sirkuit dan memiliki  $m = n - 1$  buah sisi.
- 5.  $G$  tidak mengandung sirkuit dan penambahan satu sisi pada graf akan membuat hanya satu sirkuit.
- 6.  $G$  terhubung dan semua sisinya adalah jembatan.

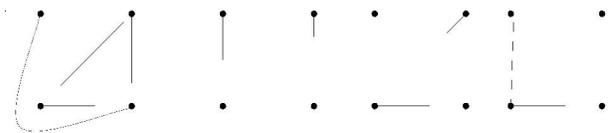
**Pohon Merentang (spanning tree)**

Suatu pohon rentangan atau spanning tree adalah suatu subgraf dari graf  $G$  yang mengandung semua simpul dari  $G$  dan merupakan suatu pohon [3]. Pohon merentang dari graf terhubung adalah upagraf merentang yang berupa

pohon. Artinya, setiap vertex terletak di pohon, tetapi tidak ada siklus (atau loop) terbentuk. Di sisi lain, setiap jembatan dari  $G$  harus milik  $T$ .

Sebuah pohon rentang dari graf terhubung  $G$  juga dapat didefinisikan sebagai satu set maksimal tepi  $G$  yang berisi siklus tidak ada, atau sebagai seperangkat minimal tepi yang menghubungkan semua titik. Dalam bidang-bidang tertentu teori grafik sering berguna untuk mencari pohon rentang minimum dari sebuah graf berbobot . masalah optimasi lain di pohon mencakup juga telah dipelajari, termasuk mencakup pohon maksimum, pohon minimum yang mencakup setidaknya simpul  $k$ , pohon rentang minimum dengan di tepi  $k$  paling per titik (Gelara-Dibatasi Spanning Tree) , pohon merentang dengan jumlah daun terbesar (erat kaitannya dengan terkecil terhubung mendominasi set) , pohon merentang dengan daun paling sedikit (berkaitan erat dengan masalah jalan Hamilton) , dengan rentang diameter pohon minimal, dan dilatasi minimum spanning tree [4].

Perhatikan graf  $G$  dibawah ini dengan beberapa pohon perentang dari graf  $G$  tersebut yaitu  $T1$ ,  $T2$ , dan  $T3$ .



**Gambar 2.1** Graf dan pohon

Setiap graf terhubung mempunyai paling sedikit satu buah pohon merentang. Graf tak-terhubung dengan  $k$  komponen mempunyai  $k$  buah hutan merentang yang disebut hutan merentang (*spanning forest*).

**KODE BINER**

Setiap angka 0 dan 1 biasa disebut **Bit**. Bit adalah singkatan dari **Binary Digit**. Kata *Binary* diambil dari nama *Binary Number System* (Sistem Bilangan Biner) [5]. Tabel berikut ini menunjukkan ukuran bit :

**Tabel 2.1** Ukuran Bit

0	1 bit
01	2 bit
0110	4 bit
10011101	8 bit

Bit hanya mengenal angka 0 dan 1, yang dalam elektornik menunjukkan adanya arus dan tidak adanya arus. Semua karakter dan data dalam elektorikan senantia dikonver dalam angka 0 dan 1. Sistem ini dikenal dengan istilah sitem bilangan biner.

**Sistem Bilangan Biner**

Sistem bilangan biner disusun dari angka-angka, sama seperti sistem bilangan desimal (sistem bilangan 10) yang sering digunakan saat ini. Tetapi untuk desimal menggunakan angka 0 sampai 9, sistem bilangan biner hanya menggunakan angka 0 dan 1 [5]. Pada tabel berikut adalah contoh perbandingan angka dalam sistem bilangan biner dan desimal.

**Tabel 2.2** Perbandingan angka desimal dan biner

Sistem Desimal	Sistem Biner
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111

Pengolahan data dalam komputer seperti dalam pengolah kata (*word processing*), maupun microsoft Excel atau lainnya, dikenal berbagai bentuk data seperti huruf, angka maupun karakter lainnya. Ketika melakukan suatu pengolahan kata, komputer bekerja dengan *keyboard*. Ada 101 tombol yang mewakili karakter alphabet A, B, C, dst. Selain itu juga akan ditemui karakter angka 0 sampai dengan 9, dan karakter-karakter lain yang diperlukan, antara lain : ,;-;():\_?!”#\*%&.

Seluruh karakter yang ada pada keyboard harus didigitalkan. Karakter-karakter tersebut diwakili oleh angka-angka 0 dan 1. Bit yang digunakan adalah 8 bit biner dinamakan *Byte*. Setiap karakter selalu direpresentasikan dalam 8 bit. 8 bit = 1 bytes. Byte inilah menjadi ukuran besarnya data dalam komputer.

Untuk sistem bilangan biner, banyaknya kombinasi dihitung dengan  $2^n$  d” m, dimana n adalah jumlah bit, m adalah kombinasi yang dapat diwakili, sehingga pada 8 bit biner, dapat mewakili  $2^8 = 256$  kombinasi maksimal [5]. Pada sistem komputer dikenal 256 karakter, mulai dari a sampai z, A sampai Z, 0 sampai 9 dan berbagai simbol lainnya, dimana setiap karakter tersebut memiliki kode ASCII dan kode biner masing masing, lebih jelasnya lihat tabel 3 dibawah ini:

**Tabel 2.3** Kode ASCII dan kode binernya

Letter	ASCII Code	Binary	Letter	ASCII Code	Binary
A	097	01100001	A	065	01000001
b	098	01100010	B	066	01000010
c	099	01100011	C	067	01000011
d	100	01100100	D	068	01000100
e	101	01100101	E	069	01000101
f	102	01100110	F	070	01000110
g	103	01100111	G	071	01000111
h	104	01101000	H	072	01001000
i	105	01101001	I	073	01001001
j	106	01101010	J	074	01001010
k	107	01101011	K	075	01001011
l	108	01101100	L	076	01001100
m	109	01101101	M	077	01001101
n	110	01101110	N	078	01001110
o	111	01101111	O	079	01001111
p	112	01110000	P	080	01010000
q	113	01110001	Q	081	01010001
r	114	01110010	R	082	01010010
s	115	01110011	S	083	01010011
t	116	01110100	T	084	01010100
u	117	01110101	U	085	01010101
v	118	01110110	V	086	01010110
w	119	01110111	W	087	01010111
x	120	01111000	X	088	01011000
y	121	01111001	Y	089	01011001
z	122	01111010	Z	090	01011010

**Algoritma Huffman**

Algoritma Huffman, yang dibuat oleh seorang mahasiswa MIT bernama David Huffman pada tahun 1951, merupakan salah satu metode paling lama dan paling terkenal dalam kompresi teks [4]. Kode Huffman dikenal sebagai kode yang digunakan untuk melakukan proses kompresi data [9]. Kode ini dibentuk dengan memanfaatkan frekuensi kemunculan karakter pada berkas yang ingin dikompresi. Kode yang diperoleh ialah berupa bilangan biner yang kemudian diubah ke dalam karakter yang bersesuaian dengan bilangan biner tersebut. Algoritma Huffman menggunakan prinsip pengkodean yang mirip dengan kode Morse, yaitu tiap karakter (simbol) dikodekan hanya dengan rangkaian beberapa bit, dimana karakter yang sering muncul dikodekan dengan rangkaian bit yang pendek dan karakter yang jarang muncul dikodekan dengan rangkaian bit yang lebih panjang.

Berdasarkan tipe peta kode yang digunakan untuk mengubah pesan awal (isi data yang diinputkan) menjadi sekumpulan *codeword*, algoritma Huffman termasuk kedalam kelas algoritma yang menggunakan metode statik . Metoda statik adalah metoda yang selalu menggunakan peta kode yang sama. Metoda ini membutuhkan dua fase (*two-pass*): fase pertama untuk menghitung probabilitas kemunculan tiap simbol dan menentukan peta kodenya, dan fase kedua untuk mengubah pesan menjadi kumpulan kode yang akan di transmisikan.

Sedangkan berdasarkan teknik pengkodean simbol yang digunakan, algoritma Huffman menggunakan metode *symbolwise*. Metode *symbolwise* adalah metode yang menghitung peluang kemunculan dari setiap simbol dalam satu waktu, dimana simbol yang lebih sering muncul diberi kode lebih pendek dibandingkan simbol yang jarang muncul.

**Pembentukan Pohon Huffman**

Kode Huffman pada dasarnya merupakan kode prefiks (*prefix code*). Kode prefiks adalah himpunan yang berisi sekumpulan kode biner, dimana pada kode prefik ini tidak ada kode biner yang menjadi awal bagi kode biner yang lain. Kode prefiks biasanya direpresentasikan sebagai pohon biner yang diberikan nilai atau label. Untuk cabang kiri pada pohon biner diberi label 0, sedangkan pada cabang kanan pada pohon biner diberi label 1. Rangkaian bit yang terbentuk pada setiap lintasan dari akar ke daun merupakan kode prefiks untuk karakter yang berpadanan. Pohon biner ini biasa disebut pohon Huffman.

Langkah-langkah pembentukan pohon Huffman adalah sebagai berikut [8] :

1. Baca semua karakter di dalam teks untuk menghitung frekuensi kemunculan setiap karakter. Setiap karakter penyusun teks dinyatakan sebagai pohon bersimpul tunggal. Setiap simpul di-assign dengan frekuensi kemunculan karakter tersebut.
2. Terapkan strategi algoritma *greedy* sebagai berikut : gabungkan dua buah pohon yang mempunyai frekuensi terkecil pada sebuah akar. Setelah digabungkan akar tersebut akan mempunyai frekuensi yang merupakan jumlah dari frekuensi dua buah pohon-pohon penyusunnya.
3. Ulangi langkah 2 sampai hanya tersisa satu buah pohon Huffman. Agar pemilihan dua pohon yang akan digabungkan berlangsung cepat, maka semua yang ada selalu terurut menaik berdasarkan frekuensi.

Misalkan sebuah pesan berisi “Selamat Datang”. Pesan ini memiliki 10 karakter dengan karakter “a” muncul paling banyak yaitu sebanyak 4 kali dan t sebanyak 2 kali, selain itu muncul sebanyak satu akali saja.

**Tabel 2.4** Kode Biner dan ASCII dari pesan “selamat datang”

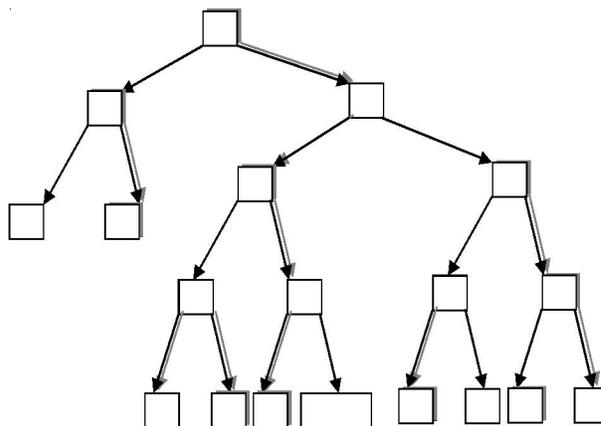
Huruf	ASCII	Biner	Frek	Panjang Bit	Total Bit
a	097	01100001	4	8	32
d	100	01100100	1	8	8
e	101	01100101	1	8	8
g	103	01100111	1	8	8
l	108	01101100	1	8	8
m	109	01101101	1	8	8
n	110	01101110	1	8	8
s	115	01110011	1	8	8
t	116	01110100	2	8	16
Spasi	32	00100000	1	8	8
Total Bit					112

*Encoding* adalah cara menyusun *string* biner dari teks yang ada. Proses *encoding* untuk satu karakter dimulai dengan membuat pohon Huffman terlebih dahulu. Setelah itu, kode untuk satu karakter dibuat dengan menyusun nama *string* biner yang dibaca dari akar sampai ke daun pohon Huffman.

Langkah-langkah untuk men-*encoding* suatu string biner adalah sebagai berikut [6]:

1. Tentukan karakter yang akan di-*encoding*
2. Mulai dari akar, baca setiap bit yang ada pada cabang yang bersesuaian sampai ketemu daun dimana karakter itu berada
3. Ulangi langkah 2 sampai seluruh karakter di-*encoding*

Dengan menggunakan algoritma Huffman di atas maka dapat dibentuk graf pohon sebagai berikut.



**Gambar 2.2** Graf Pohon dari pesan “selamat datang”

Pada tabel distribusi kemunculan huruf pada pesan diatas menunjukkan bahwa huruf a yang paling banyak muncul demikian juga dengan huruf t sehingga pada graf pohon menjadi satu cabang tersendiri. Cabang ini memiliki lintasan yang paling pendek dibanding dengan lintasan lainnya. Akibatnya huruf a maupun t memiliki kode Huffman yang paling pendek dibanding huruf lainnya. Berikut tabel lengkap kode Huffman berdasarkan graf pohon yang dibentuk pada gambar 1 diatas.

**Tabel 2.5** Kode Huffman dari pesan “selamat datang”

Huruf	Bit	Panjang Bit
A	01	2
T	00	2
G	1000	4
M	1001	4
L	1010	4
Spasi	1011	4
S	1100	4
E	1101	4
D	1110	4
N	1111	4

Dengan frekuensi kemunculan huruf pada pesan diatas maka jumlah bit pada hasil kode Huffman dengan graf pohon diatas dapat mengikuti tabel berikut: