

BAB III

MASALAH HUFFMAN

(MENENTUKAN SANDI DARI SUATU PERINTAH)

Definisi 15 :

Kode yang dibentuk dari angka binary yang hanya memerlukan dua angka yaitu 0 dan 1 dinamakan kode binary.

Definisi 16 :

Simbol huruf adalah huruf dari A sampai dengan Z.

Definisi 17 :

Rangkaian dari simbol huruf yang masing-masing simbol huruf penyusunnya akan dicari kode binary-nya, dinamakan perintah.

Disamping menentukan kode binary dari simbol huruf, dalam bab ini akan disajikan pula penentuan perangkaian kode binary dari suatu perintah yang bertujuan untuk menghasilkan penulisan sandi.

Definisi 18 :

Rangkaian kode binary yang menyajikan suatu perintah merupakan sandi.

Dalam penulisan sandi, panjang sandi tergantung dari panjang masing-masing kode binary dari simbol huruf yang penyusunnya. Cara menentukan kode binary dari suatu simbol huruf tidak harus mempunyai bit yang sama panjang, akan tetapi dapat ditentukan dengan cara yang akan dibahas selanjutnya pada bab ini. Hanya saja panjang sandi diusahakan seminim mungkin. Dengan sendirinya bit

dari kode diusahakan seminim mungkin juga.

Masalah Huffman merupakan penyajian dari suatu binary tree. Masalah ini mempunyai tujuan untuk menulis suatu perintah yang berupa huruf-huruf atau suatu perintah yang terdiri dari simbol-simbol dalam suatu rangkaian kode binary. Rangkaian kode binary tersebut bertujuan untuk menghasilkan penulisan sandi.

Sebagai contoh, andaikan diketahui 4 simbol huruf A, B, C dan D dan pemberian kode simbol huruf ini seperti dibawah ini :

Simbol	Kode
A	010
B	100
C	000
D	111

Maka suatu perintah : ABACCDA

sandinya : 010100010000000111010

panjang sandi : 21 bit

Karena menggunakan 3 bit untuk masing-masing simbol huruf, maka untuk contoh diatas diperlukan 21 bit untuk penulisan suatu perintah dalam sandi. Maka dicari kemungkinan lain agar penulisan sandi lebih efisien dalam penggunaan bit. Andaikan simbol huruf A,B,C,D diberi kode 2 bit seperti dibawah ini :

Simbol	Kode
A	00
B	01
C	10
D	11

maka untuk perintah : ABCCDA
 sandinya : 00010010101100
 panjang sandi : 14 bit

Panjang sandi 14 bit memang lebih pendek daripada panjang sandi 21 bit, tetapi bukanlah panjang sandi minimal untuk perintah ABACCDA. Jika kode binary untuk simbol huruf A ditentukan dengan bit yang lebih pendek dari simbol huruf B dan D, maka panjang sandi menjadi minimal, karena kode binary yang panjang jarang muncul.

Definisi 20 :

Kode dari simbol huruf yang jarang muncul diberi bit yang lebih panjang dari simbol huruf yang sering muncul.

Definisi 21 :

- Panjang sandi ditentukan oleh banyaknya bit yang dipergunakan untuk menulisnya.
- Panjang sandi dikatakan minimal apabila untuk suatu perintah yang sama, bit yang dipergunakan paling pendek.

3.1. CARA MENENTUKAN KODE BINARY DARI SIMBOL HURUF SUATU PERINTAH

Sesuai dengan definisi 18., maka bit kode binary dari simbol huruf menentukan panjang sandi dari suatu perintah.

Definisi 22 :

Frekwensi suatu simbol huruf adalah banyaknya simbol huruf itu muncul dalam suatu perintah.

Definisi 23 :

Dua simbol huruf yang digabungkan akan mempunyai frekwensi baru yang merupakan jumlah frekwensi masing-masing simbol huruf.

Langkah 1 :

Menemukan simbol huruf yang paling jarang muncul dalam suatu perintah.

Apabila terdapat dua simbol huruf, langsung ditentukan kode binarynya dengan angka 0 dan angka 1 untuk masing-masing simbol huruf.

Akan tetapi, apabila lebih dari dua simbol huruf diusahakan penggabungan dua-dua simbol huruf, karena angka yang dipergunakan hanya 0 dan 1 sesuai dengan Definisi 15.

Bit terakhir dari kode binarynya akan membedakan keduanya.

Pada langkah ini dicari simbol huruf yang paling jarang muncul sesuai dengan Definisi 20.

Karena bit yang akan dicari adalah bit yang terakhir dari kode, padahal untuk langkah selanjutnya bit kode tersebut akan terus bertambah dengan angka 0 atau 1 sampai semua simbol huruf yang membentuk perintah termuat.

Langkah 2 :

Kombinasikan kedua simbol huruf pada langkah 1 dalam simbol tunggal yang menyajikan pengertian bahwa simbol huruf tersebut adalah salah satu dari simbol huruf yang dikombinasikan.

Langkah 3:

Sesuai dengan Definisi 23 maka untuk langkah 2 telah ditentukan frekwensi baru.

Temukan lagi simbol huruf yang mempunyai frekwensi terkecil.

Bit terakhir dari kodenya akan membedakan keduanya.

Angka 0 dan angka 1 ditentukan lagi.

Ket : Angka 1 (untuk simbol gabungan) apabila simbol gabungan hanya ada satu, apabila lebih ditentukan sendiri simbol huruf gabungan mana yang diberi angka 0 atau 1.

Langkah 4:

Kombinasikan lagi kedua simbol huruf pada langkah 3 dalam simbol tunggal.

Demikianlah, langkah-langkah di atas dilakukan berulang kali sampai semua simbol huruf yang membentuk suatu perintah termuat dan hal ini menunjukkan angka 0 sebagai kodenya. Sekarang kode binary tree masing-masing simbol huruf dapat ditentukan dari kanan ke kiri karena yang mula-mula dicari adalah bit yang terakhir.

Contoh :

Diketahui : Suatu perintah ABACCDA

Ditanyakan : Kode binary simbol huruf yang membentuk perintah dan sandinya.

Penyelesaian :

1. Menentukan langkah 1, ditemukan dua huruf yaitu, huruf B dan D. Untuk B diberi angka 0 dan untuk D diberi angka 1.

2. Menggunakan langkah 2, simbol huruf tersebut menjadi BD.
3. Menggunakan langkah 3, frekwensi baru BD adalah 2 dan frekwensi terkecil lainnya adalah C. Bit terakhir kode BD dan C tersebut akan membedakan keduanya. Untuk C diberi angka 0 untuk BD diberi angka 1. (Dengan demikian, simbol huruf B dan D masing-masing kodenya sudah 2 bit).
4. Kombinasikan C dan BD dalam simbol huruf tunggal menjadi CBD dan frekwensi baru CBD adalah 4 (lihat Definisi 23).
5. Dicari lagi frekwensi yang terkecil, yaitu A dengan frekwensi 3. Bit terakhir dari A dan CBD akan membedakan keduanya. Untuk A diberi angka 0 dan untuk CBD diberi angka 1.
6. Kombinasikan A dan CBD dalam simbol tunggal ACBD. Jadi simbol tersebut sudah memuat semua simbol huruf yang memuat suatu perintah. Kode binary dari masing-masing simbol huruf dapat ditentukan :

* Simbol ACBD terdiri dari simbol A dan simbol CBD

- untuk kode 0 simbol hurufnya A

- untuk kode 1 simbol hurufnya C, B atau D

* Simbol CBD terdiri dari simbol C dan BD

- untuk 10 simbol hurufnya C

- untuk 11 simbol hurufnya B atau D

Bit pertama menunjukkan bahwa simbol adalah salah satu pembentuk dari CBD dan bit kedua menunjukkan apakah simbol huruf C atau BD.

* Simbol BD terdiri dari simbol B dan D

- untuk 110 simbol hurufnya B
- untuk 111 simbol hurufnya D

Apabila dibuat daftar kode, adalah sebagai berikut:

Simbol	Kode
A	0
B	110
C	10
D	111

jadi untuk suatu perintah : ABACCCA

sandinya : 0110010101110

panjang sandi : 13 bit

Jika 0 ditentukan sebagai bit pertama, simbolnya adalah A, jika sebaliknya (bukan 0) simbolnya adalah B,C atau D dan bit berikutnya diteliti lagi. Jika bit kedua adalah 0 maka simbolnya adalah C, sebaliknya harus simbol B atau D dan bit berikutnya diteliti lagi. Jika bit ketiga adalah 0, simbol huruf adalah B dan jika 1 simbolnya D. Setelah simbol huruf yang pertama diketahui, proses diulang lagi, dimulai dari bit sesudahnya untuk menemukan simbol kedua.

3.2. MENENTUKAN KODE MENGGUNAKAN BINARY TREE

Sebenarnya penggabungan dua simbol huruf menjadi satu merupakan penggunaan suatu binary tree. Karena setiap vertex dalam binary tree selalu mempunyai dua son (son kiri dan son kanan). Untuk menentukan kode dengan binary tree simbol huruf asli disajikan dengan leaf, sedangkan simbol huruf gabungan disajikan dengan vertex bukan leaf.

Definisi 24 :

Pada suatu binary tree, pada kedua edge yang insiden dari masing-masing cabang diberi angka 0 dan angka 1. Angka 0 untuk cabang kiri dan angka 1 untuk cabang kanan.

Definisi 25 :

Pada setiap leaf dari binary tree, dipasangkan suatu rangkaian yaitu rangkaian angka edge-edge pada lintasan yang menghubungkan root dengan leaf tersebut dan rangkaian tersebut merupakan kode dari simbol huruf pada leaf itu.

Disini diinginkan untuk menyandikan simbol huruf/menentukan kode simbol huruf sebagai rangkaian binary, sedemikian sehingga setiap simbol huruf mempunyai kode yang tunggal.

Teorema 13 :

Rangkaian binary dari simbol huruf - simbol huruf (kode) mempunyai panjang maksimal 5 bit dan dapat dengan tunggal menyandikan simbol huruf tersebut.

Bukti :

Menurut Teorema 10 jika banyak vertex suatu binary tree berlevel k adalah n, maka berlaku :

$$n \leq 2^0 + 2^1 + 2^2 + \dots + 2^k.$$

Simbol huruf banyaknya ada 26 maka $2^4 < 26 < 2^5$.

Jadi panjang maksimal 5 bit. Adapun dengan tunggal

menyandikan simbol huruf tersebut dibuktikan sebagai berikut : Simbol huruf disajikan oleh leaf dari binary tree dan bila leaf tersebut berada pada

cabang yang berbeda (definisi 24), maka angkanya akan berbeda. Dengan demikian rangkaiannya akan berbeda untuk masing-masing leaf.

Dalam suatu perintah, simbol huruf yang membentuknya tidak merata frekwensi penggunaannya. Huruf a dan i akan lebih sering digunakan daripada simbol huruf x dan v.

Masalah yang kemudian timbul adalah kemungkinan adanya kode yang membingungkan penerima berita.

Misalnya :

- Simbol huruf A mempunyai kode 00
- Simbol huruf U mempunyai kode 01
- Simbol huruf Q mempunyai kode 0001

Maka apabila penerima berita menerima sandi 0001, maka dia akan bingung mengenai perintah tadi, karena dapat berarti AU atau Q.

Definisi 26 :

Kode dari suatu binary tree adalah rangkaian binary, dimana tidak ada anggotanya yang merupakan kode dari salah satu anggota yang lain.

Contoh :

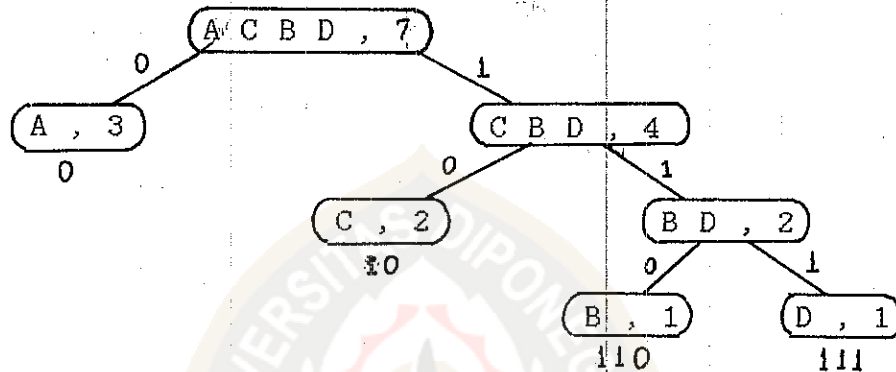
{ 000,001,01,11} adalah kode dari suatu binary tree, sedangkan {1,00,01,000,0001} bukan kode karena 00 merupakan kode awal dari 01 dan 000 merupakan kode awal dari 1. Seperti telah disinggung pada bagian sebelumnya pengga-

bungkan dua simbol huruf menjadi tunggal merupakan penggabungan binary tree. Dengan demikian, untuk suatu perintah dapat dicari kodenya dengan binary tree sebagai berikut:

- Menggabungkan frekwensi terkecil secara bertahap

- Pemberian angka sesuai definisi 24
- Pemasangan rangkaian binary sesuai definisi 25

Dapat dikonstruksikan binary tree seperti di bawah ini untuk perintah ABACCCA :



Gambar 6

Keterangan Gambar 6 :

- Masing-masing vertex memuat simbol huruf dan frekwensinya
- Masing-masing leaf menunjukkan simbol huruf asli
- Masing-masing bukan leaf menunjukkan simbol huruf gabungan
- Rangkaian binary di bawah leaf menunjukkan kode dari simbol huruf

3.3 MENGKONSTRUKSIKAN SUATU BINARY TREE DARI KODE

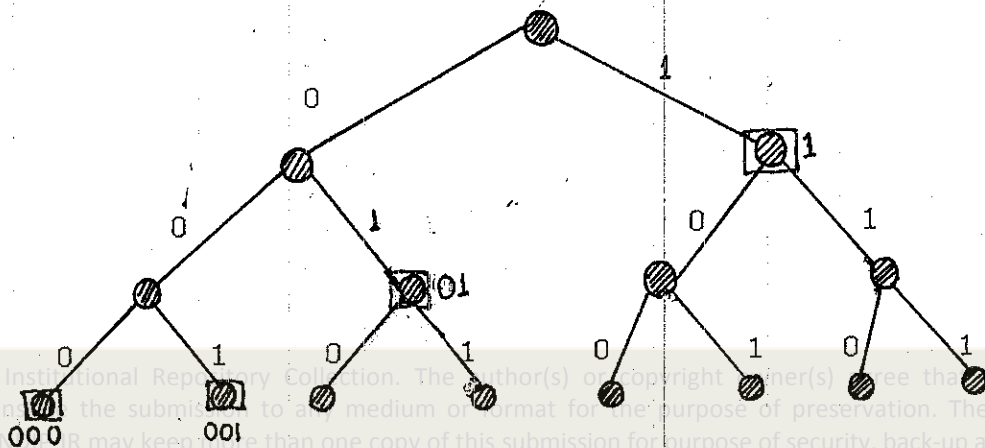
Pada bagian sebelumnya telah disajikan, bagaimana menentukan kode suatu binary tree. Disini akan disajikan cara mengkonstruksikan suatu binary tree dari kode yang telah diketahui, sebagai berikut :

1. Misalkan h adalah panjang rangkaian terpanjang (bit terpanjang) dari kode yang diketahui. (Hal ini dimaksudkan untuk mendapat level dari binary tree).

2. Konstruksikan binary tree regular penuh dengan tinggi h dan setiap dua edge yang insiden pada suatu cabang, diberi angka 0 dan 1.
3. Pada setiap vertex, tentukan barisan angka yaitu yang berada dalam lintasan yang menghubungkan vertex tersebut dengan root.
4. Kemudian kode yang sudah diketahui dicocokkan, setiap vertex yang cocok dengan salah satu anggota dari kode diberi tanda.
5. Hapus vertex-vertex yang tidak bertanda beserta descendantnya yang juga tak bertanda. Atau dengan perkataan lain, subtree dengan vertex itu sebagai root dihapus, maka yang tinggal adalah binary tree yang berkorespondensi dengan kode yang diberikan.

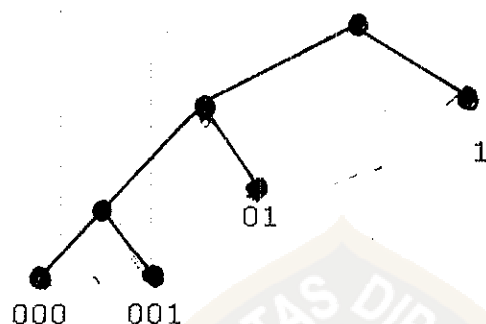
Contoh :

Misalkan diketahui kode binary $\{1, 000, 01, 001\}$. Maka $h = 3$, dan dikonstruksikan binary tree regular penuh dengan tinggi 3 sebagai berikut :



(http://Gambar 7.ac.id)

Setelah itu, subtree dengan vertex yang tidak bertanda sebagai root dihapus, dan gambar 8 di bawah ini adalah binary tree yang berkorespondensi dengan kode-kode yang dimaksud.



Gambar 8

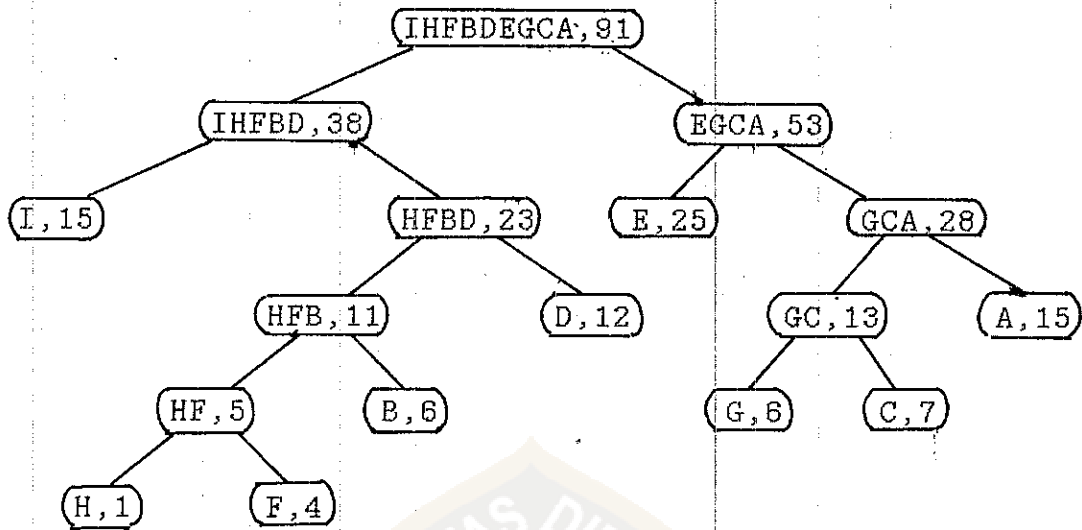
3.4 HUFFMAN TREE

Diketahui suatu daftar yang berisi simbol huruf dan frekwensi sebagai berikut :

Simbol	frekwensi
A	15
B	6
C	7
D	12
E	25
F	4
G	6
H	1
I	15

Dengan menggunakan cara yang sama pada bagian yang terdahulu pada bab ini, maka binary treenya oleh Huffman

dikonstruksikan sebagai berikut :



Gambar 9 : Huffman Tree

Pertama kali Huffman tree dibuat, kode dari simbol huruf manapun dapat ditentukan. Membuat tree dan menghasilkan kode hanya diperlukan hubungan masing-masing vertex ke fathernya dan apakah vertex-vertex tersebut son kiri atau son kanan. Hubungan father dengan selain kedua son tadi tidak diperlukan.

Kode suatu leaf dari Huffman Tree, dapat ditentukan dengan 2 (dua) cara :

1. Dimulai dari leaf tree yang menyajikan simbol huruf itu dan naik ke root. Ketentuannya adalah, setiap kali suatu cabang kiri naik, suatu nilai/angka 0 ditentukan pada sebelah kiri kode dan setiap kali cabang kanan naik, nilai /angka 1 ditentukan pada sebelah kiri kode.
2. Dimulai dari root tree, setiap angka 0 ditemukan, cabang kiri diturunkan dan setiap angka 1 ditentukan, cabang kanan diturunkan. Proses ini diulangi terus sampai suatu leaf ditemukan. Kode ditemukan dengan menghubungkan cabang kiri atau cabang kanan

ke leaf tersebut.

Dengan demikian, gambar 9 yang merupakan Huffman Tree, masing-masing leaf yang menyajikan simbol huruf, dapat ditentukan kode-kodenya sebagai berikut :

Simbol	kode
A	111
B	0101
C	1101
D	011
E	10
F	01001
G	1100
H	01000
I	00

Menggunakan simbol huruf yang sudah diketahui kodenya yang ditemukan oleh Huffman yang dinamakan Huffman Tree dapat dilihat gambar 9 , maka sandi :

1110100010111011 dapat dibaca A HEAD.

Setiap vertex yang bukan leaf dari Huffman Tree, mempunyai sub tree kanan dan sub tree kiri yang tidak kosong. Tree seperti ini dinamakan strictly binary tree. Strictly binary tree dengan p leaf mempunyai $2p - 1$ vertex. Leaf dari Huffman Tree menyajikan simbol huruf, maka jumlah vertex dapat dihitung dari jumlah simbol huruf. Jika ada p simbol huruf, maka vertex Huffman tree adalah $2p - 1$.