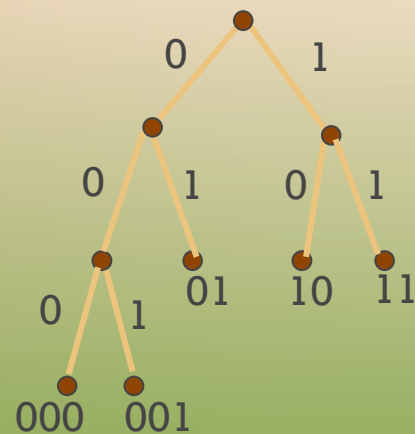


PERTEMUAN 15

- KODE AWALAN
- KODE HUFFMAN
- TRAVERSAL POHON BINER

KODE AWALAN

- Definisi himpunan kode (misalnya kode biner) sedemikian hingga tidak ada anggota kumpulan yang merupakan awalan dari anggota yang lain
- Mempunyai pohon biner yang bersesuaian
- Sisi diberi label 0 atau 1, semua sisi kiri diberi label 0 saja (atau 1 saja) sedangkan sisi kanan diberi label 1 saja (atau 0 saja)
- Barisan sisi-sisi yang dilalui oleh lintasan dari akar ke daun menyatakan kode awalan (ditulis di daun)
- Kegunaan untuk :
 - Mengirim pesan pada komunikasi data
 - Setiap karakter di dalam pesan direpresentasikan dengan barisan angka 0 dan 1
 - Untuk pembentukan kode Huffman dalam pemampatan data (data compression)



Pohon biner dari kode awalan (000, 001, 01, 10, 11)

KODE HUFFMAN

- Pemampatan data dilakukan dengan mengkodekan setiap karakter di dalam pesan atau di dalam arsip dikodekan dengan kode yang lebih pendek
- Sistem kode yang banyak digunakan adalah kode ASCII (setiap karakter dikodekan dalam 8 bit biner)
- Cara pembentukan kode Huffman dengan membentuk pohon biner (dinamakan dengan pohon Huffman) yaitu :
 1. Pilih 2 simbol dengan peluang (probability) paling kecil sebagai child kemudian kedua simbol tersebut dikombinasikan sebagai parent peluang penjumlahan dari kedua simbol tersebut
 2. Pilih 2 simbol berikutnya termasuk simbol baru yang mempunyai peluang kecil sebagai child kemudian kedua simbol tersebut dikombinasikan sebagai parent peluang penjumlahan dari kedua simbol tersebut
 3. Prosedur yang sama dilakukan pada 2 simbol berikutnya yang mempunyai peluang terkecil sebagai child kemudian kedua simbol tersebut dikombinasikan sebagai parent peluang penjumlahan dari kedua simbol tersebut
- Kode Huffman tidak bersifat unik, artinya kode untuk setiap karakter berbeda-beda pada setiap pesan bergantung pada kekerapan kemunculan karakter tersebut di dalam pesan
- Keputusan apakah suatu simpul pada pohon Huffman diletakkan di kiri atau di kanan menentukan kode yang dihasilkan (tetapi tidak mempengaruhi panjang kodenya)

KODE HUFFMAN

Representasikan string 'ABACCDA' dalam kode ASCII dan kode Huffman

Kode ASCII

Simbol	Kode ASCII
A	01000001
B	01000010
C	01000011
D	01000100

String 'ABACCDA' direpresintasikan menjadi rangkaian :

01000001 01000010 01000001 01000011 01000011 01000100 01000001
A B A C C D A

Representasi 7 huruf tersebut membutuhkan $7 \times 8 \text{ bit} = 56 \text{ bit}$ (7 byte)

KODE HUFFMAN

Kode Huffman

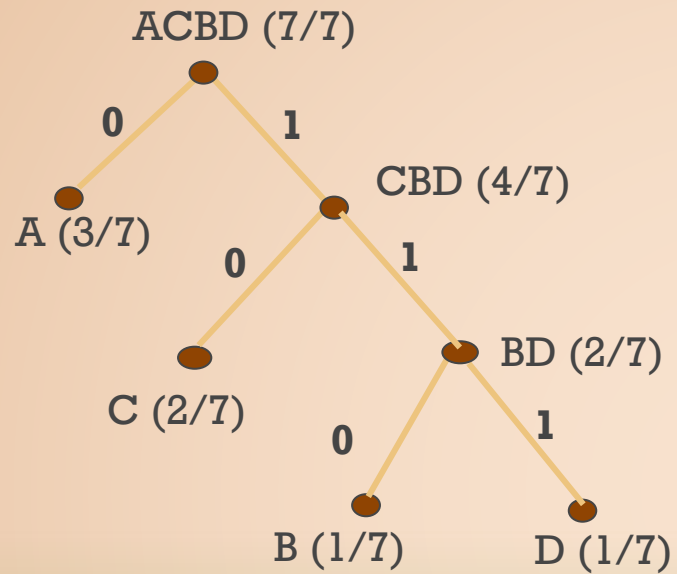
Simbol	Kekerapan	Peluang
A	3	$3/7$
B	1	$1/7$
C	2	$2/7$
D	1	$1/7$

Cara pembentukan kode Huffman :

1. Pilih 2 simbol dengan peluang paling kecil, yaitu simbol B dan D. Simbol tersebut dikombinasikan menjadi simbol BD dengan peluang $1/7 + 1/7 = 2/7$
2. Pilih 2 simbol dengan peluang paling kecil, yaitu simbol C dan BD. Simbol tersebut dikombinasikan menjadi simbol CBD dengan peluang $2/7 + 2/7 = 4/7$
3. Pilih 2 simbol dengan peluang paling kecil, yaitu simbol A dan CBD. Simbol tersebut dikombinasikan menjadi simbol ACBD dengan peluang $4/7 + 3/7 = 7/7 = 1$

KODE HUFFMAN

Pohon Huffman untuk pesan 'ABACCDA'



String 'ABACCDA' direpresintasikan menjadi rangkaian :

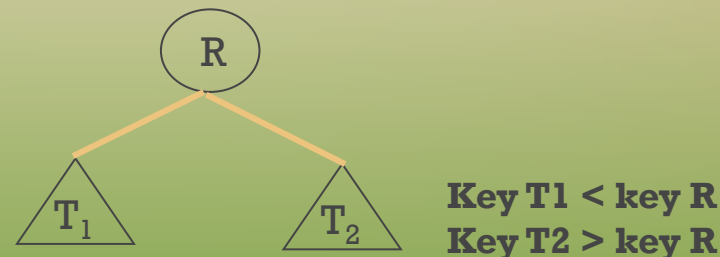
0110010101110

Representasi 7 huruf tersebut membutuhkan 13 bit

Simbol	Kekerapan	Peluang	Kode Huffman
A	3	3/7	0
B	1	1/7	110
C	2	2/7	10
D	1	1/7	111

POHON PENCARIAN BINER

- Definisi pohon biner yang setiap key diatur dalam suatu urutan tertentu
- Digunakan untuk melakukan operasi
 - Pencarian
 - Penyisipan
 - Penghapusan elemen
- Simpul pada pohon pencarian berupa field kunci (key) pada :
 - Data record *atau*
 - Data itu sendiri
- Key (kunci) adalah :
 - Nilai yang membedakan setiap simpul dengan simpul yang lainnya
- Key harus unik, karena itu tidak ada 2 buah simpul atau lebih yang mempunyai kunci yang sama
- Jika R adalah akar dan semua key yang tersimpan pada setiap simpul tidak ada yang sama maka : :
 - Semua simpul pada subpohon kiri mempunyai key lebih kecil dari key R
 - Semua simpul di subpohon kanan mempunyai key nilai lebih besar dari key R



TRAVERSAL POHON BINER

- Misalkan T adalah pohon biner, akarnya R , subpohon kiri T_1 dan subpohon kanan T_2 maka ada skema mengunjungi simpul-simpul di dalam pohon biner T :
 1. Preorder
 - Kunjungi R (sekaligus memproses simpul R)
 - Kunjungi T_1 secara preorder
 - Kunjungi T_2 secara preorder
 2. Inorder
 - Kunjungi T_1 secara inorder
 - Kunjungi R (sekaligus memproses simpul R)
 - Kunjungi T_2 secara inorder
 3. Postorder
 - Kunjungi T_1 secara postorder
 - Kunjungi T_2 secara postorder
 - Kunjungi R (sekaligus memproses simpul R)
- Proses yang dilakukan terhadap simpul yang dikunjungi misalnya
 - mencetak informasi yang disimpan di dalam simpul
 - Memanipulasi nilai
 - Dan lain-lain

SKEMA MENGGUNJUNGI POHON BINER

PREORDER



Langkah 1 : Kunjungi R

**Langkah 2 : Kunjungi
T₁ secara preorder**

**Langkah 3 : Kunjungi
T₂ secara preorder**

INORDER



Langkah 2 : Kunjungi R

**Langkah 1 : Kunjungi
T₁ secara inorder**

**Langkah 3 : Kunjungi
T₂ secara inorder**

POSTORDER



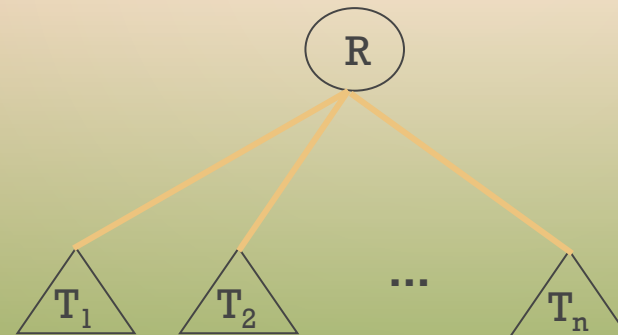
Langkah 3 : Kunjungi R

**Langkah 1 : Kunjungi
T₁ secara postorder**

**Langkah 2 : Kunjungi
T₂ secara postorder**

PENULUSURAN POHON M-ARY

- Preorder :
 - Kunjungi R
 - Kunjungi T_1, T_2, \dots, T_n secara preorder
- Inorder :
 - Kunjungi T_1 secara inorder
 - Kunjungi R
 - Kunjungi T_2, T_3, \dots, T_n secara inorder
- Postorder :
 - Kunjungi T_1, T_2, \dots, T_n secara postorder
 - Kunjungi R



Skema pohon m-ary

1. Rinaldi Munir. (2016). Matematika Diskrit. Bandung : Penerbit Informatika
2. Diktat dan Handout Matematika Diskrit. Tim Dosen Universitas Indraprasta PGRI .