



# Pohon

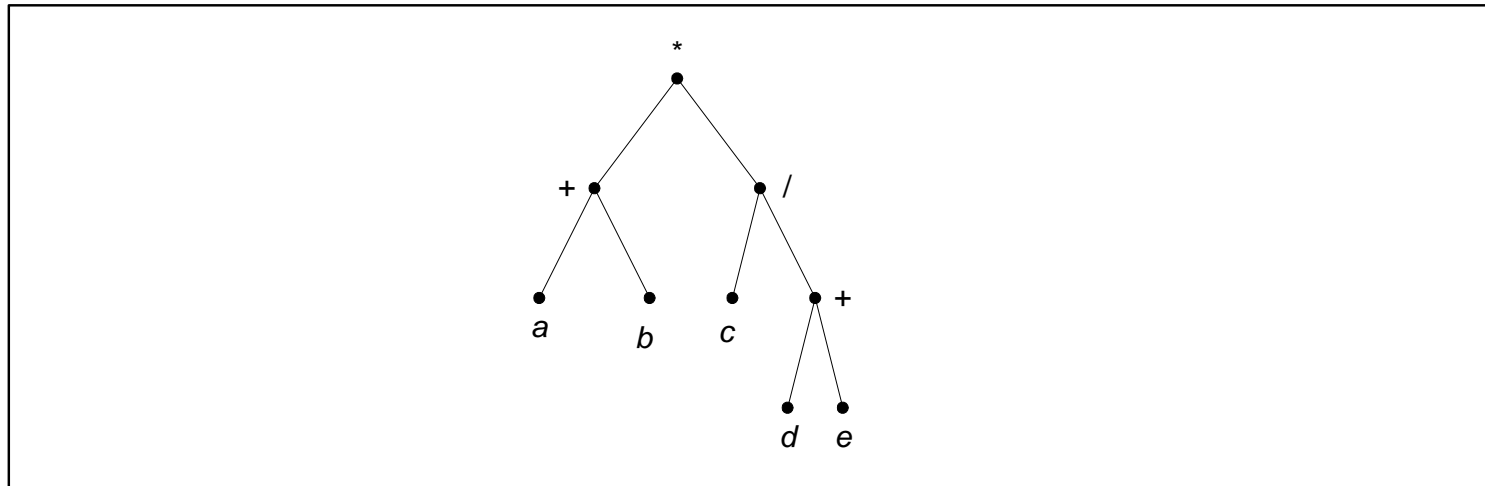
---

Bekerjasama dengan

*Rinaldi Munir*

# Terapan Pohon Biner

## 1. Pohon Ekspresi



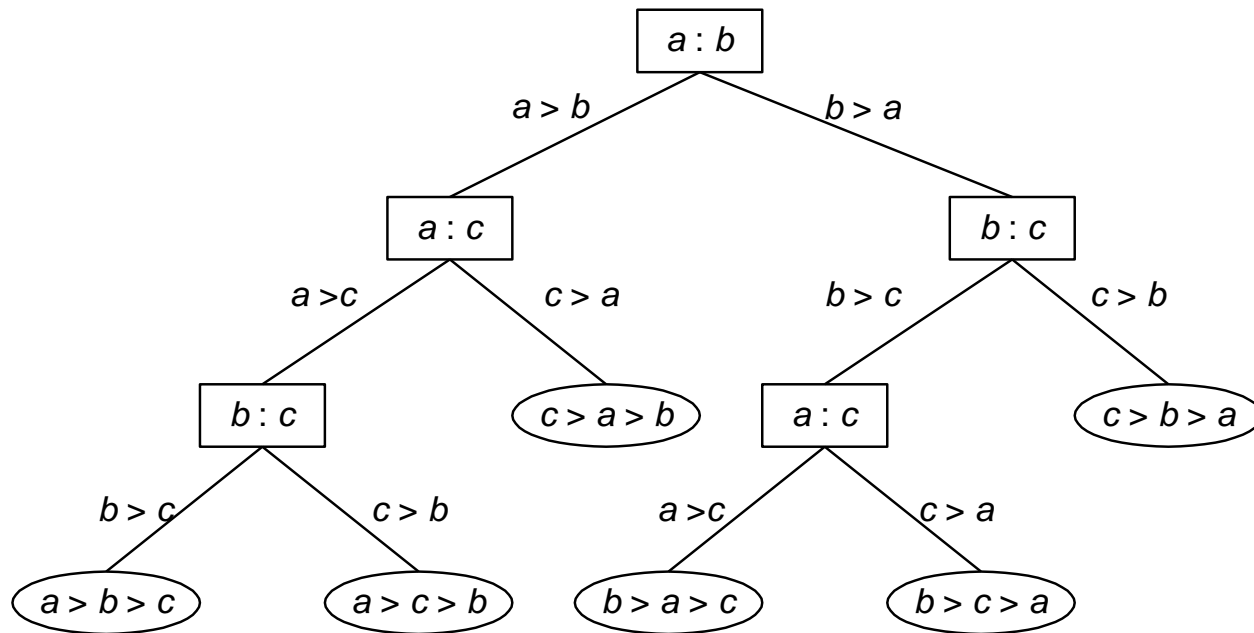
**Gambar** Pohon ekspresi dari  $(a + b) * (c / (d + e))$

daun  $\rightarrow$  *operand*

simpul dalam  $\rightarrow$  operator



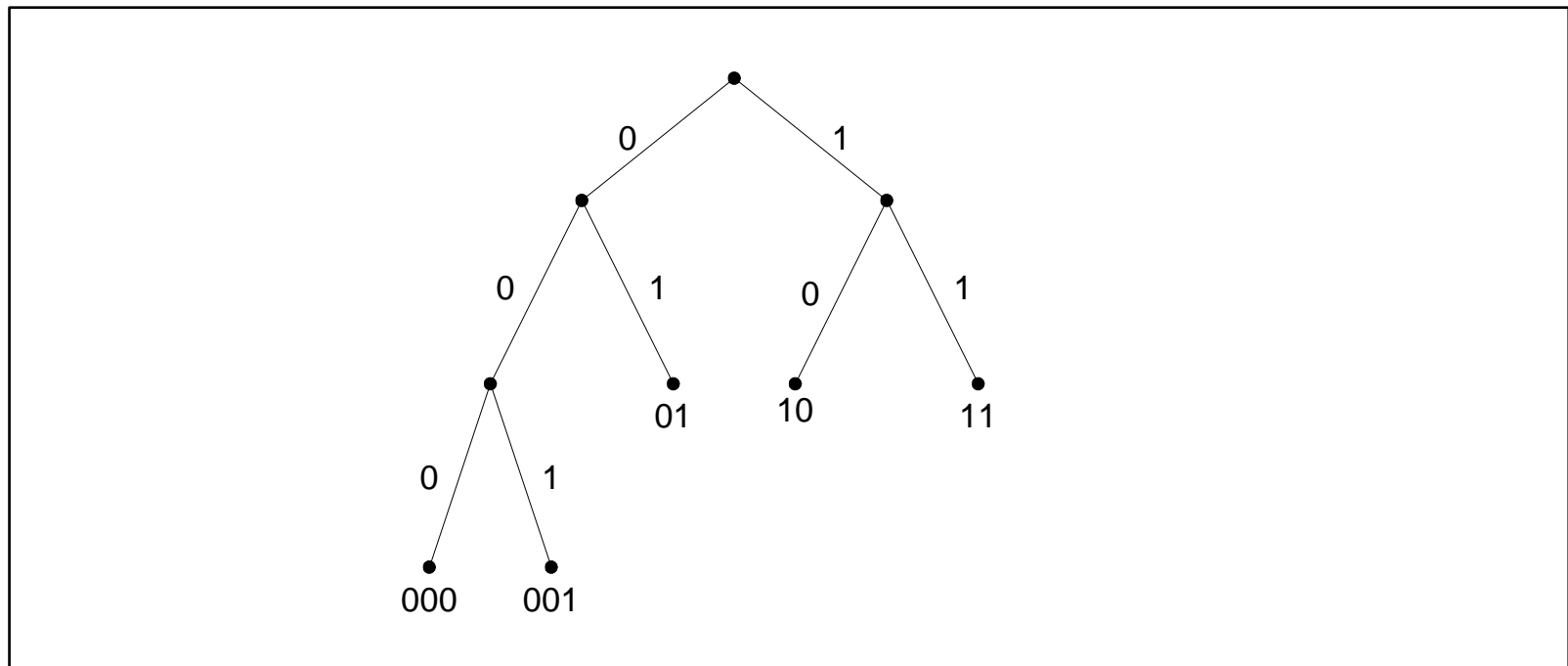
## 2. Pohon Keputusan



**Gambar** Pohon keputusan untuk mengurutkan 3 buah elemen



### 3. Kode Awalan



**Gambar** Pohon biner dari kode prefiks { 000, 001, 01, 10, 11 }



## 4. Kode Huffman

**Tabel Kode ASCII**

Simbol	Kode ASCII
<i>A</i>	01000001
<i>B</i>	01000010
<i>C</i>	01000011
<i>D</i>	01000100

rangkaian bit untuk string '*ABACCD A*':

01000001010000010010000010100000110100000110100010001000001

atau  $7 \times 8 = 56$  bit (*7 byte*).



**Tabel** Tabel kekerapan (frekuensi) dan kode Huffman  
untuk *string* *ABACCD*A

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

Dengan kode Huffman, rangkaian bit untuk '*ABACCD*A':

0110010101110

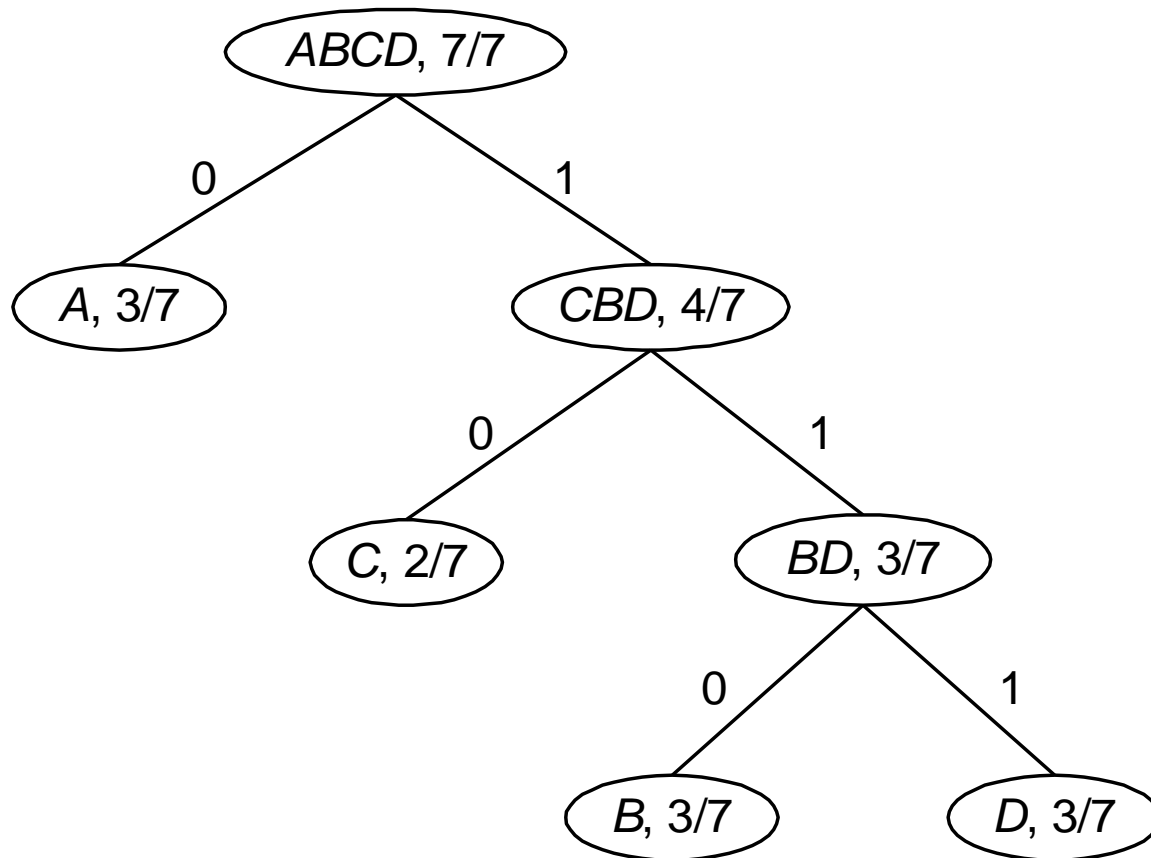
hanya 13 bit!



- Algoritma pembentukan pohon Huffman
  1. Pilih dua simbol dengan peluang (*probability*) paling kecil (pada contoh di atas simbol *B* dan *D*). Kedua simbol tadi dikombinasikan sebagai simpul orangtua dari simbol *B* dan *D* sehingga menjadi simbol *BD* dengan peluang  $1/7 + 1/7 = 2/7$ , yaitu jumlah peluang kedua anaknya.
  2. Selanjutnya, pilih dua simbol berikutnya, termasuk simbol baru, yang mempunyai peluang terkecil.
  3. Ulangi langkah 1 dan 2 sampai seluruh simbol habis.

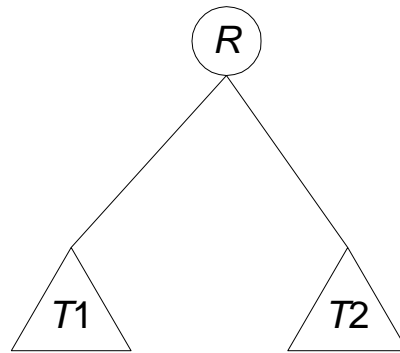


- $A = 0, C = 10, B = 110, D = 111$





## 5. Pohon Pencarian Biner

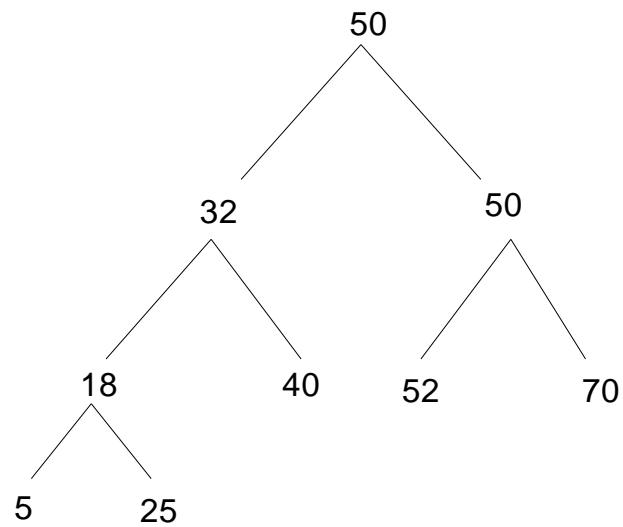


$\text{Kunci}(T1) < \text{Kunci}(R)$

$\text{Kunci}(T2) > \text{Kunci}(R)$



Data: 50, 32, 18, 40, 60, 52, 5, 25, 70



# Penelusuran (traversal) Pohon Biner

## 1. *Preorder* : $R, T_1, T_2$

- kunjungi  $R$
- kunjungi  $T_1$  secara *preorder*
- kunjungi  $T_2$  secara *preorder*

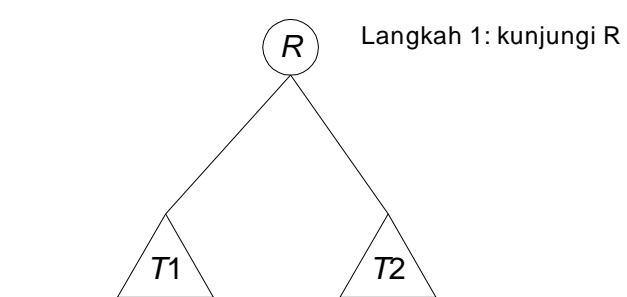
## 2. *Inorder* : $T_1, R, T_2$

- kunjungi  $T_1$  secara *inorder*
- kunjungi  $R$
- kunjungi  $T_2$  secara *inorder*

## 3. *Postorder* : $T_1, T_2, R$

- kunjungi  $T_1$  secara *postorder*
- kunjungi  $T_2$  secara *postorder*
- kunjungi  $R$

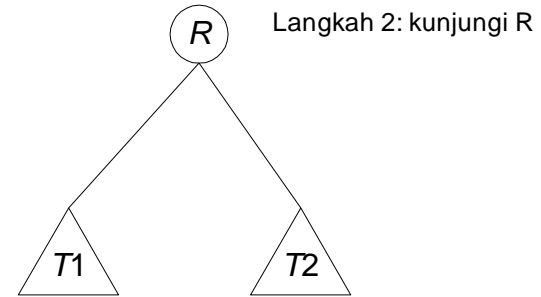




Langkah 2: kunjungi T1  
secara *preorder*

Langkah 3: kunjungi T2  
secara *preorder*

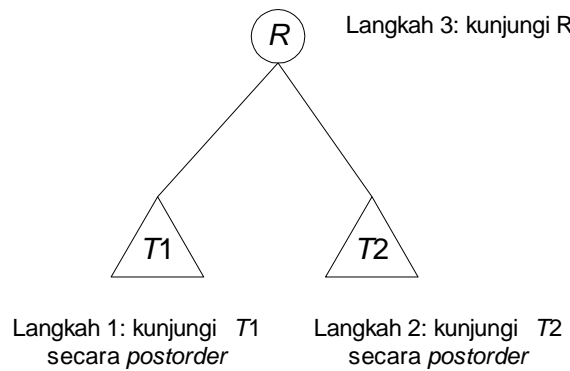
(a) *preorder*



Langkah 1: kunjungi T1  
secara *inorder*

Langkah 3: kunjungi T2  
secara *inorder*

(b) *inorder*



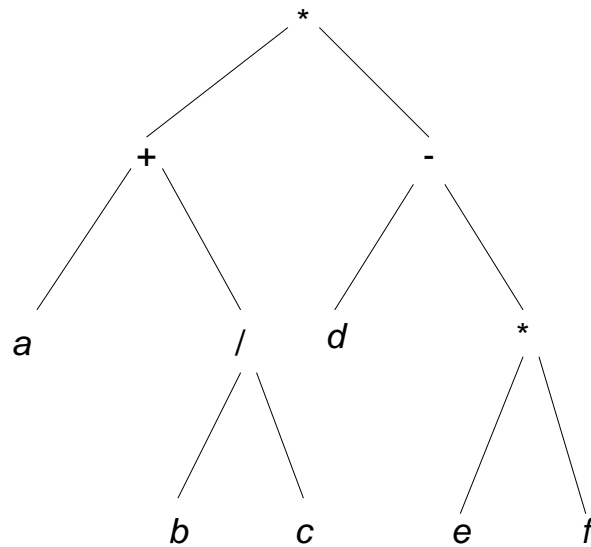
Langkah 1: kunjungi T1  
secara *postorder*

Langkah 2: kunjungi T2  
secara *postorder*

(c) *postorder*



<i>preorder</i>	: $* + a / b \ c - d * e f$	( <i>prefix</i> )
<i>inorder</i>	: $a + b / c * d - e * f$	( <i>infix</i> )
<i>postorder</i>	: $a \ b \ c / + \ d \ e \ f * - *$	( <i>postfix</i> )

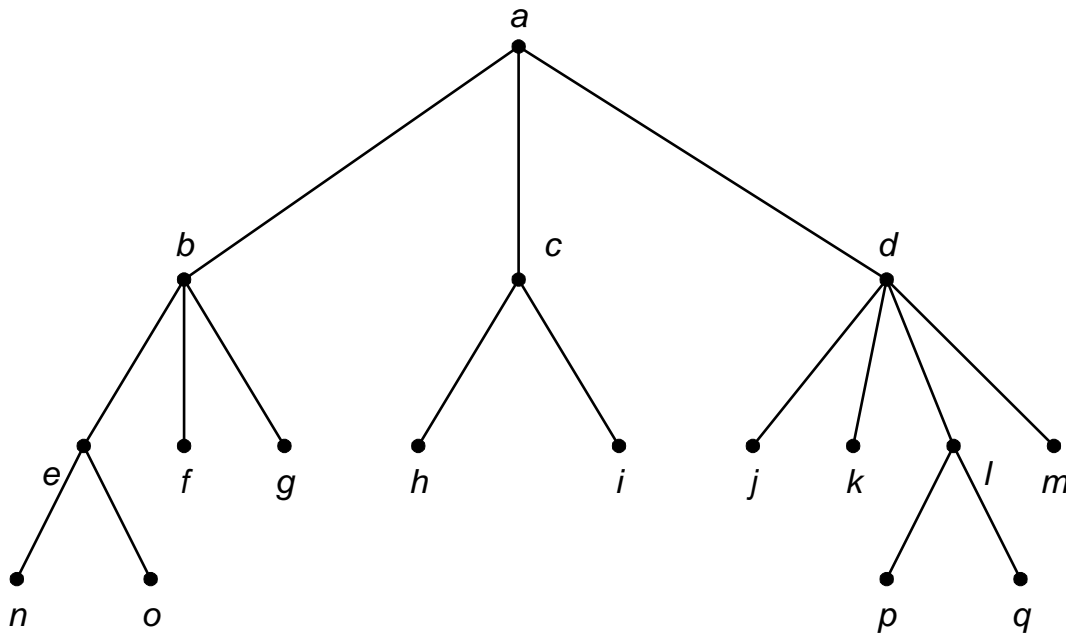


# Soal latihan

1. Diketahui 8 buah koin uang logam. Satu dari delapan koin itu ternyata palsu. Koin yang palsu mungkin lebih ringan atau lebih berat daripada koin yang asli. Misalkan tersedia sebuah timbangan neraca yang sangat teliti. Buatlah pohon keputusan untuk mencari uang palsu dengan cara menimbang paling banyak hanya 3 kali saja.



2. Tentukan hasil kunjungan *preorder*, *inorder*, dan *postorder* pada pohon 4-ary berikut ini:

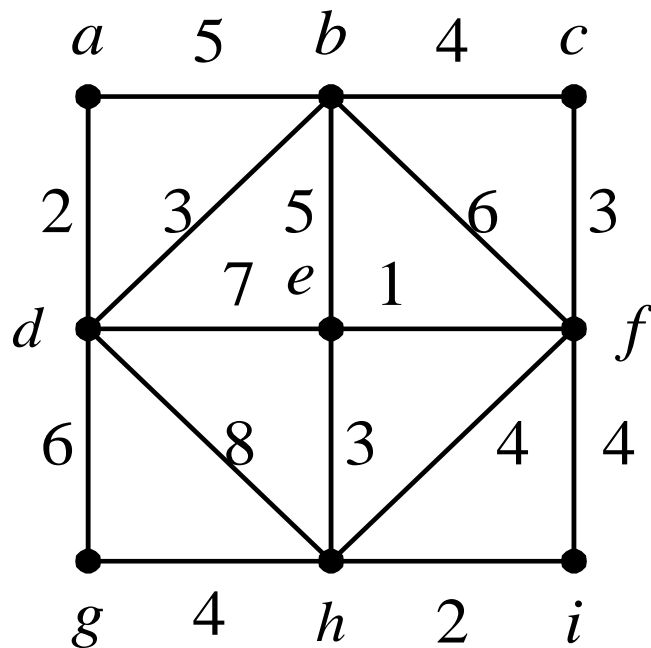


3. Gunakan pohon berakar untuk menggambarkan semua kemungkinan hasil dari pertandingan tenis antara dua orang pemain, Anton dan Budi, yang dalam hal ini pemenangnya adalah pemain yang pertama memenangkan dua set berturut-turut atau pemain yang pertama memenangkan total tiga set.





4. Tentukan dan gambarkan pohon merentang minimum dari graf di bawah ini (tahapan pembentukannya tidak perlu ditulis).



6. Diberikan masukan berupa rangkaian karakter dengan urutan sebagai berikut:

*P, T, B, F, H, K, N, S, A, U, M, I, D, C, W, O*

- (a) Gambarkan pohon pencarian (*search tree*) yang terbentuk.
- (b) Tentukan hasil penelusuran *preorder*, *inorder*, dan *postorder*, dari pohon jawaban (a) di atas.

