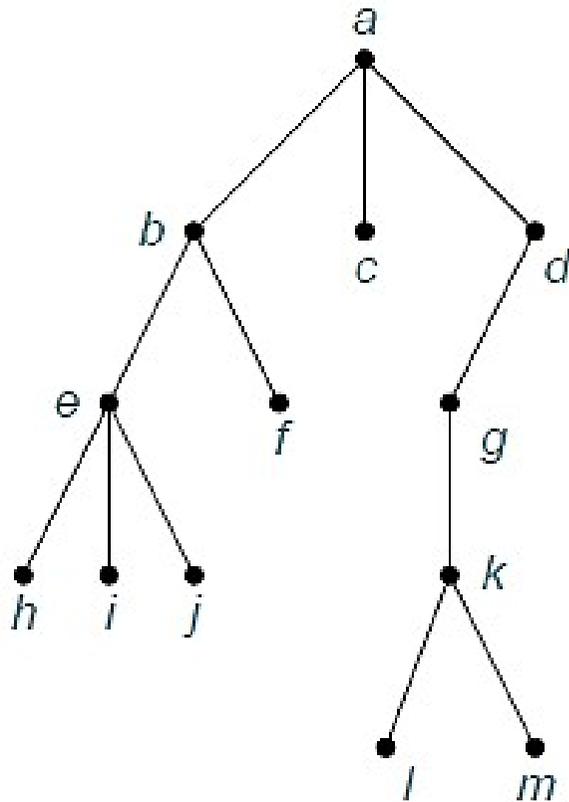


Pohon Berakar



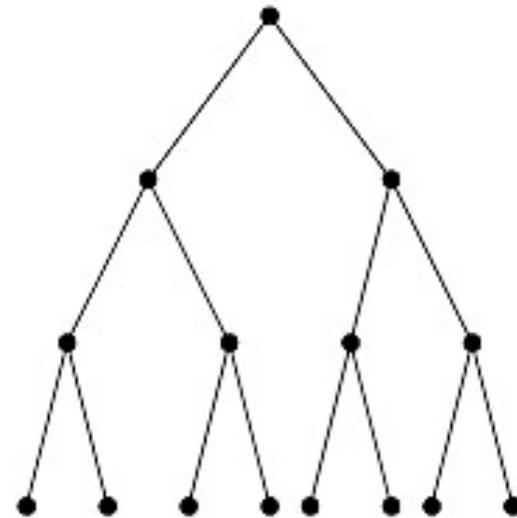
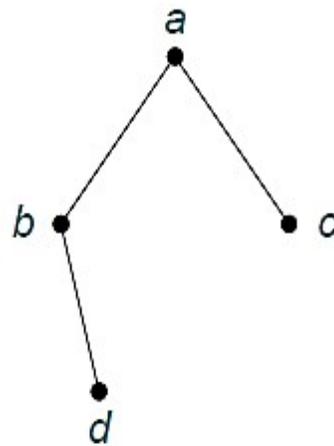
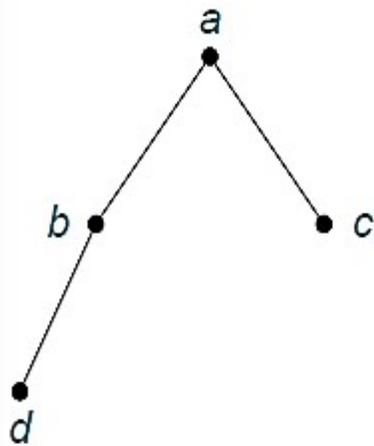
a disebut sebagai orangtua (parent), dan b, c, d merupakan anak (child/children) dari a.

Titik yang tidak mempunyai anak disebut daun. h, i, j, l, m, f, c merupakan daun.

Titik yang mempunyai anak disebut titik dalam (internal node). b, e, d, g, dan k merupakan titik dalam.

Pohon Biner

Setiap titik dalam pohon biner mempunyai paling banyak 2 buah anak



Pohon Ekspresi

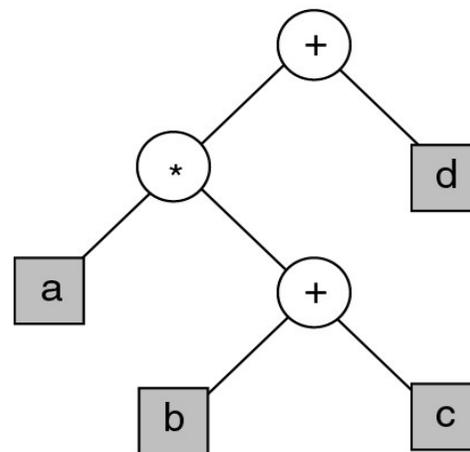
Sebuah *pohon ekspresi* adalah sebuah pohon biner dengan sifat :

Setiap daun adalah sebuah **operand**.

internal nodes adalah **operators**.

Contoh : Pohon ekspresi dari

$a * (b + c) + d$



tidak semua perancang komputer membuat algoritma untuk melakukan perhitungan data matematis seperti yang dilakukan oleh manusia pada umumnya (secara *infix*). Ada komputer yang mengolahnya secara *prefix* dan ada yang secara *postfix*.

Contoh bentuk *infix* : $A + B$

prefix : $+ A B$

postfix : $A B +$

Lambang yang digunakan manusia untuk melakukan penjumlahan angka 5 dan 7, adalah $5 + 7$ (*infix*).

5 dan 7 disebut *operand*, dan + disebut *operator*.

Untuk komputer yang melakukan proses matematis secara *prefix*, kalimatnya diubah dengan “tambahkan 5 dengan 7” atau dilambangkan dengan $+ 5 7$, dan untuk *postfix*, kalimatnya adalah “5 dan 7 ditambahkan”, atau dilambangkan dengan $5 7 +$.

Bagaimana penulisan notasi *infix* : $A + B * C + D$
secara *prefix* dan *postfix* ?.

Prefix :

langkah 1

Tentukan mana yang pertama kali yang akan diproses,
diperoleh $B * C$. Misal $B * C$ diganti dengan E, maka soalnya
menjadi :

$$A + \underline{B * C} + D = A + E + D$$

langkah 2

tentukan kembali mana yang akan diproses pertama kali. Bila

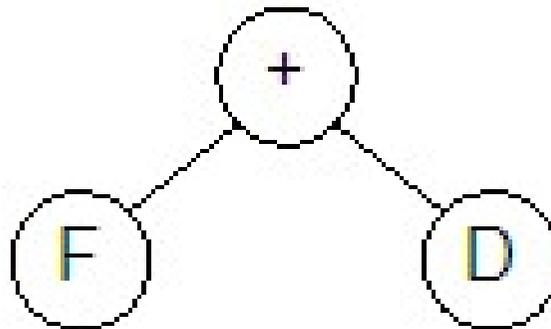
$A + E$ diganti dengan F , maka menjadi

$$\underline{A + E} + D = F + D$$

langkah 3

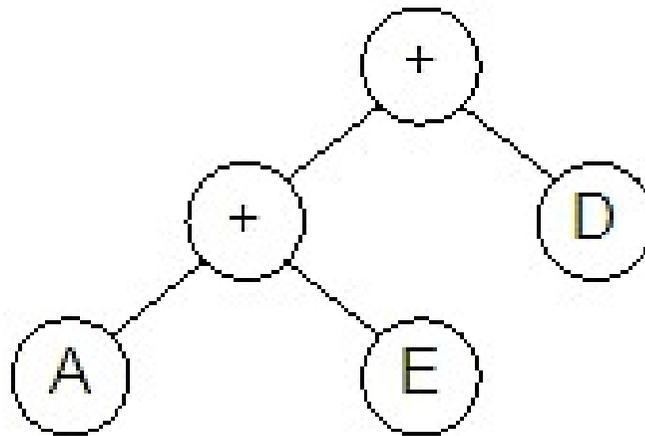
karena sudah menjadi 2 *operand*, maka notasi *prefixnya*

menjadi $+ F D$



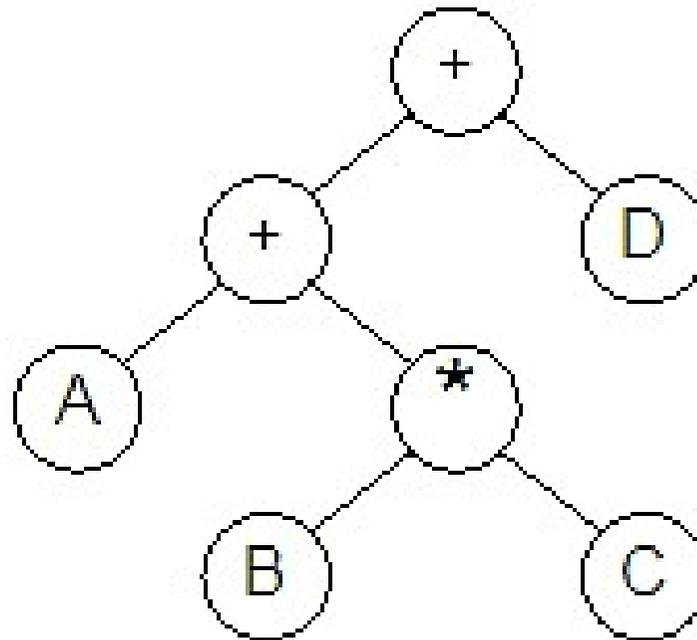
langkah 4

kembalikan nilai F sebenarnya. $F = A + E$, *prefixnya* = + A E,
sehingga notasi keseluruhannya menjadi + + A E D



langkah 5

kembalikan nilai E sebenarnya. $E = B * C$, *prefixnya* $* B C$,
sehingga notasi keseluruhannya menjadi $++ A * B C D$



Postfix :

langkah 1

tentukan mana yang pertama kali yang akan diproses, diperoleh $B * C$.

Misal $B * C$ diganti dengan E, maka soalnya menjadi : $A + E + D$

langkah 2

tentukan kembali mana yang akan diproses pertama kali, misal $A + E$.

Bila $A + E$ diganti dengan F, maka menjadi $F + D$

langkah 3

karena sudah menjadi 2 *operand*, maka notasi *postfix*nya menjadi

$F D +$

langkah 4

kembalikan nilai F. karena $F = A + E$, sehingga *postfixnya* $A E +$,
sehingga notasi keseluruhannya menjadi $A E + D +$

langkah 5

kembalikan nilai E. karena $E = B * C$, sehingga *postfixnya* $B C *$,
sehingga notasi keseluruhannya menjadi **$A B C * + D +$**