

tReE



Tenia Wahyuningrum, S.Kom. MT

Sisilia Thya Safitri, S.T., M.T

Tree ...

Kumpulan node yang **saling**
terhubung satu sama lain
dalam suatu kesatuan yang
membentuk layaknya struktur sebuah
pohon.



Tree ...

- merepresentasikan suatu struktur hirarki (**one-to-many**)
- tampak sebagai kumpulan node dari

atas ke bawah.



Tree ...

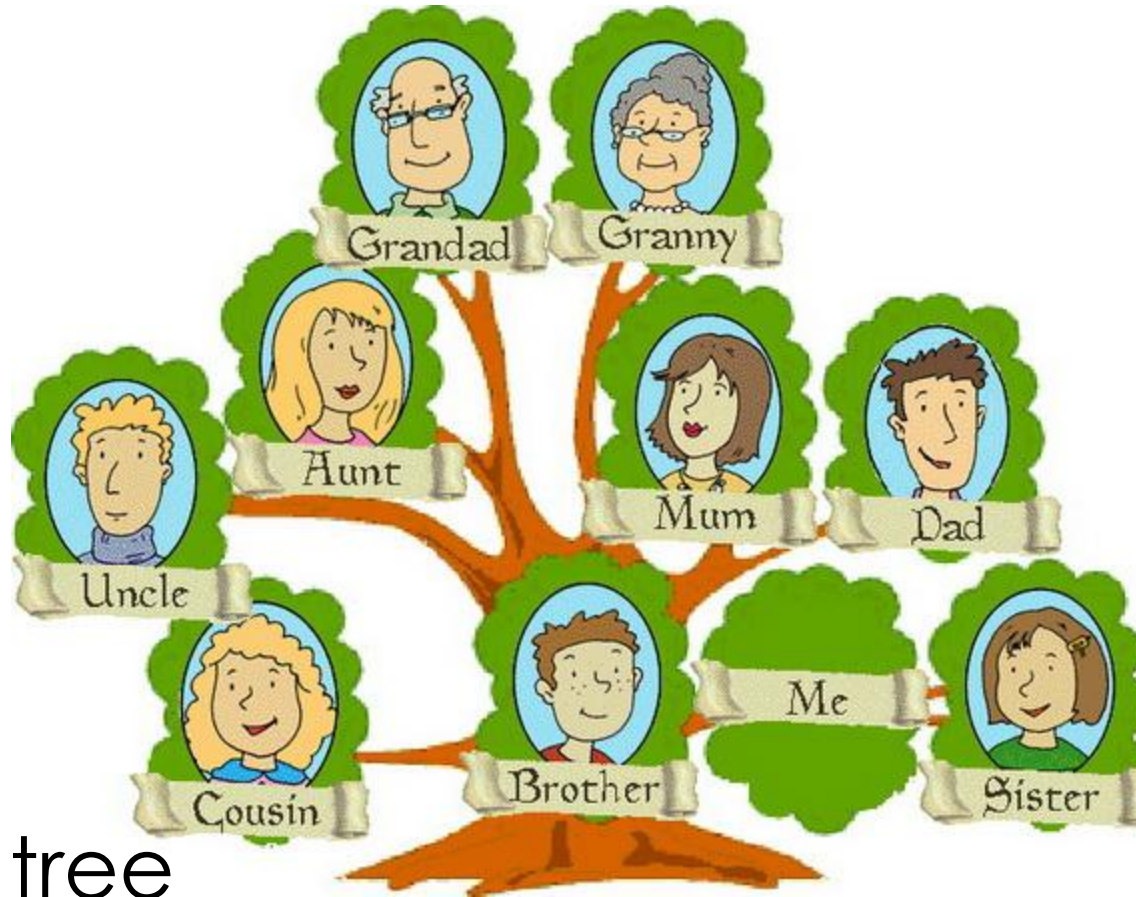
- salah satu bentuk implementasi banyak **linked list** yang digunakan untuk menggambarkan hubungan yang bersifat hirarkis





Karena harapan lah, kita menanam pohon
Meskipun kita tahu tak akan memetik buahnya berpuluh-puluh tahun kemudian

Contoh penggunaan struktur **tree**



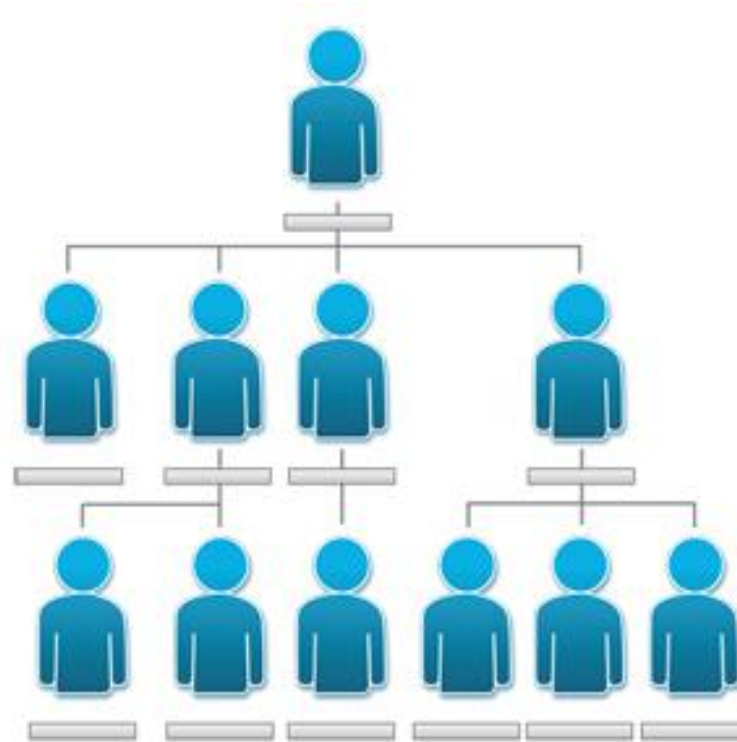
Family tree

Contoh penggunaan struktur tree



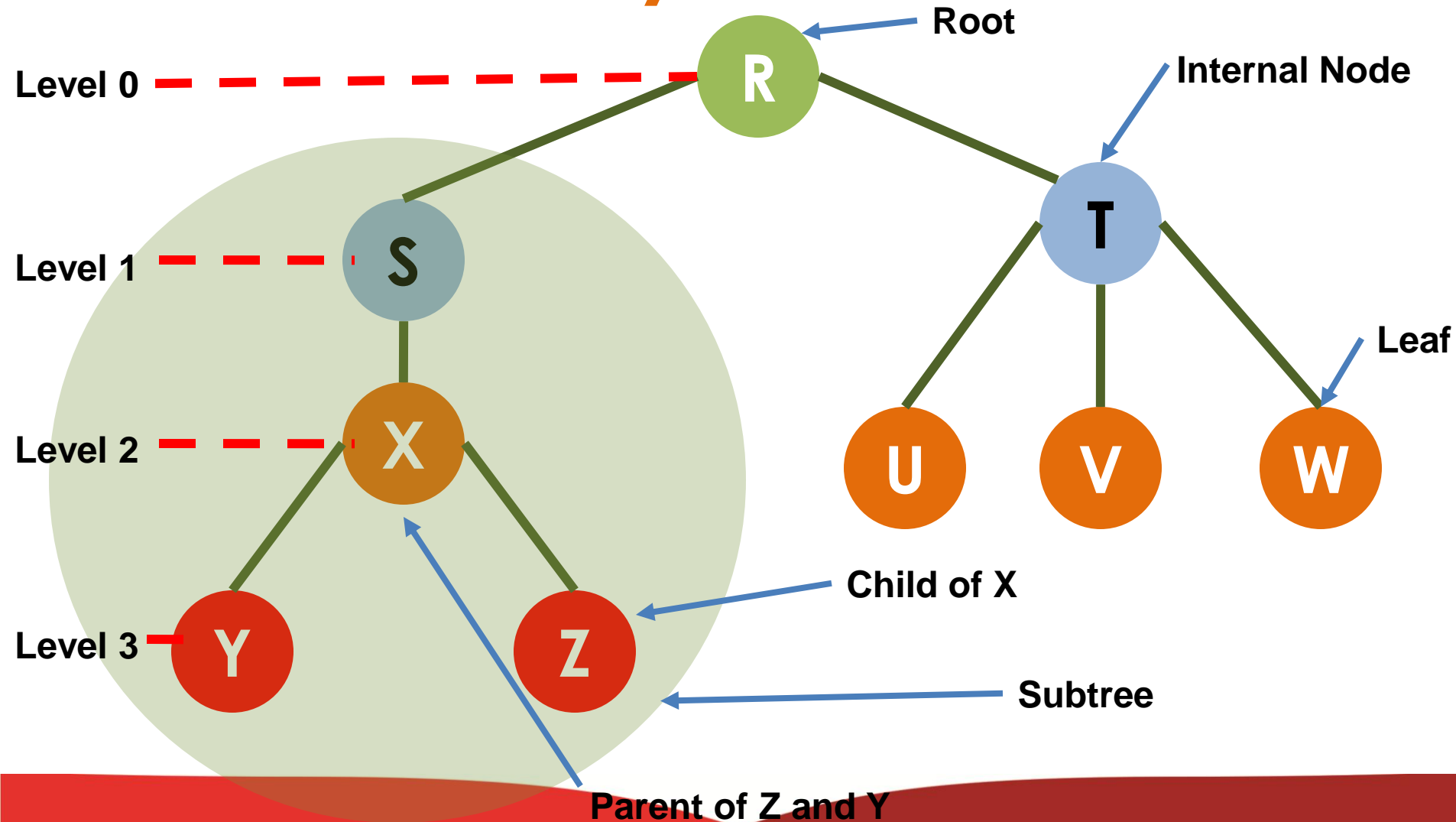
Tournamen schedule

Contoh penggunaan struktur tree



Organization Structure

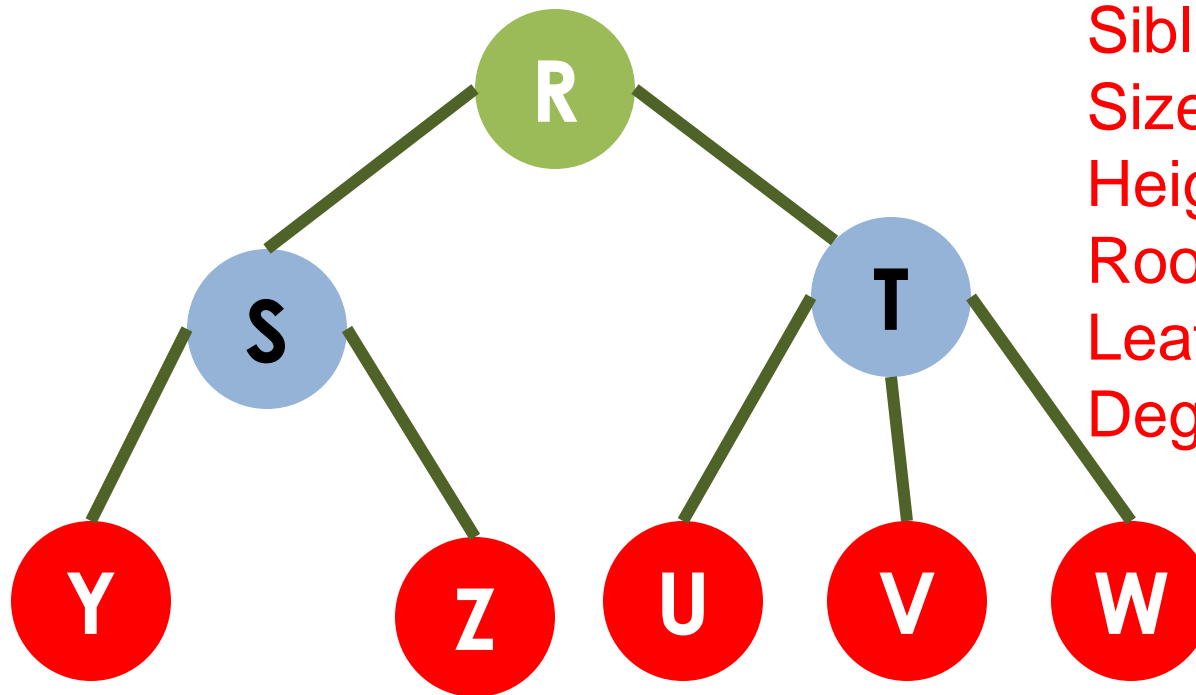
Tree anatomy



Istilah dalam Tree

Predecessor	Node yang berada diatas node tertentu.
Successor	Node yang berada dibawah node tertentu.
Ancestor	Seluruh node yang terletak sebelum node tertentu dan terletak pada jalur yang sama
Descendant	Seluruh node yang terletak setelah node tertentu dan terletak pada jalur yang sama
Parent	Predecessor satu level di atas suatu node.
Child	Successor satu level di bawah suatu node.
Sibling	Node-node yang memiliki parent yang sama
Subtree	Suatu node beserta descendantnya.
Size	Banyaknya node dalam suatu tree
Height	Banyaknya tingkatan dalam suatu tree
Root	Node khusus yang tidak memiliki predecessor.
Leaf	Node-node dalam tree yang tidak memiliki successor.
Degree	Banyaknya child dalam suatu node

a Tree



Ancestor (U) = T, R

Descendant (T) = U, V, W

Parent (Y) = S

Child (R) = S, T

Sibling (U) = V, W

Size = 8

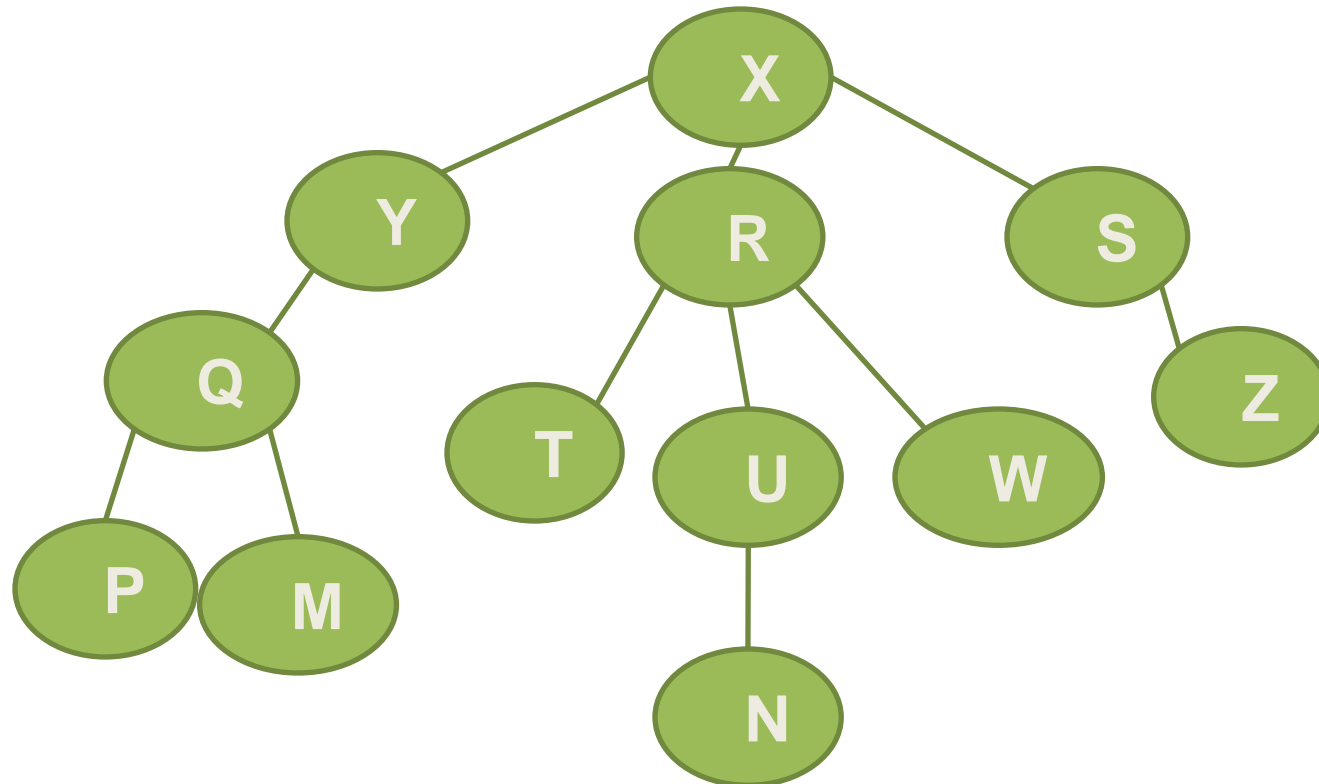
Height = 3

Root = R

Leaf = Y, Z, U, V, W

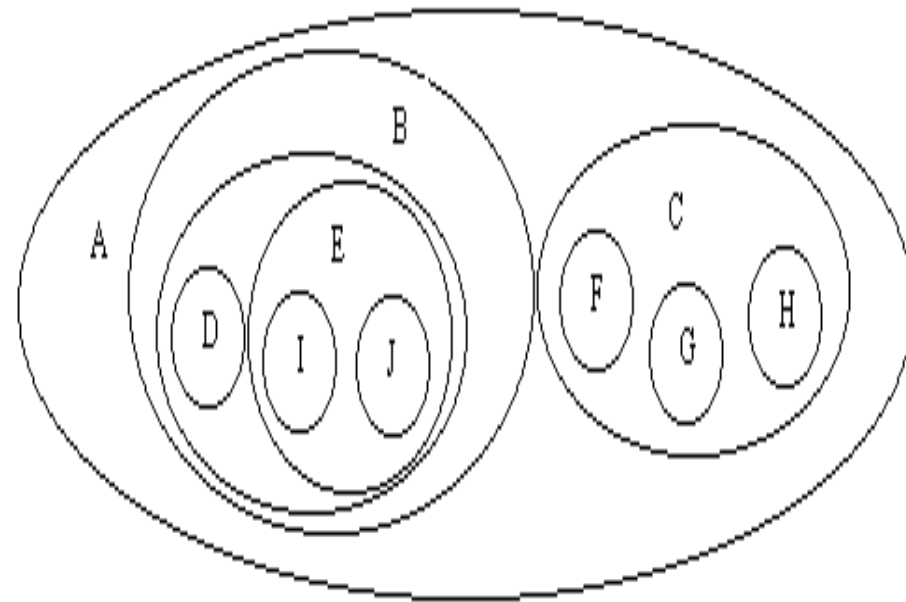
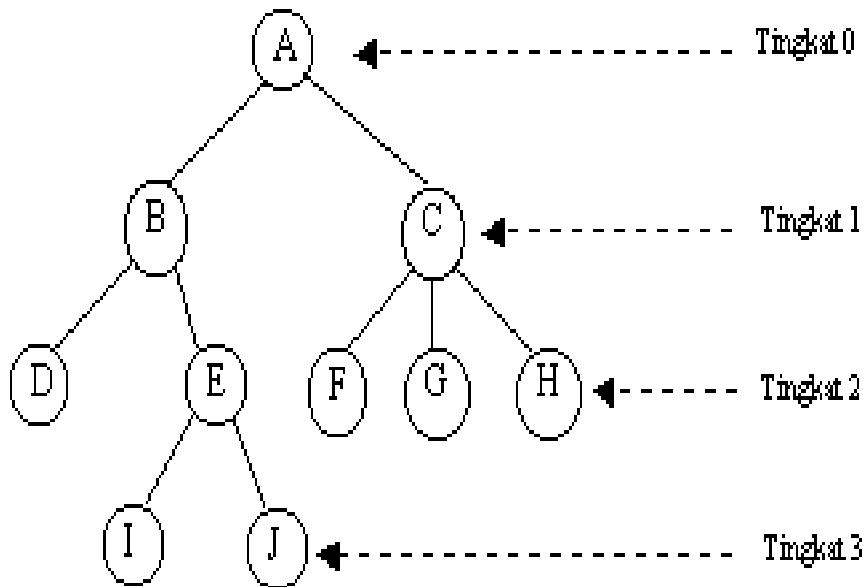
Degree (T) = 3

Latihan



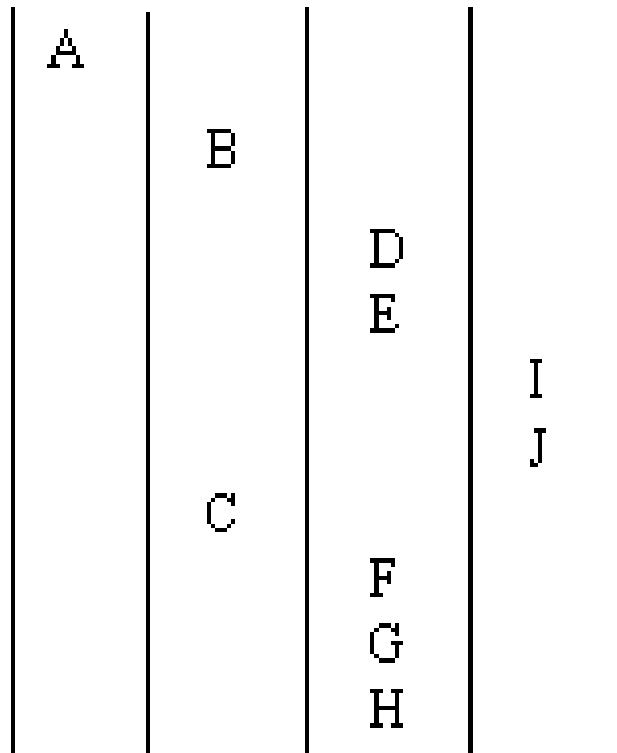
Representasi Tree

■ Diagram Venn



Tree Representation

▪ Notasi Tingkat



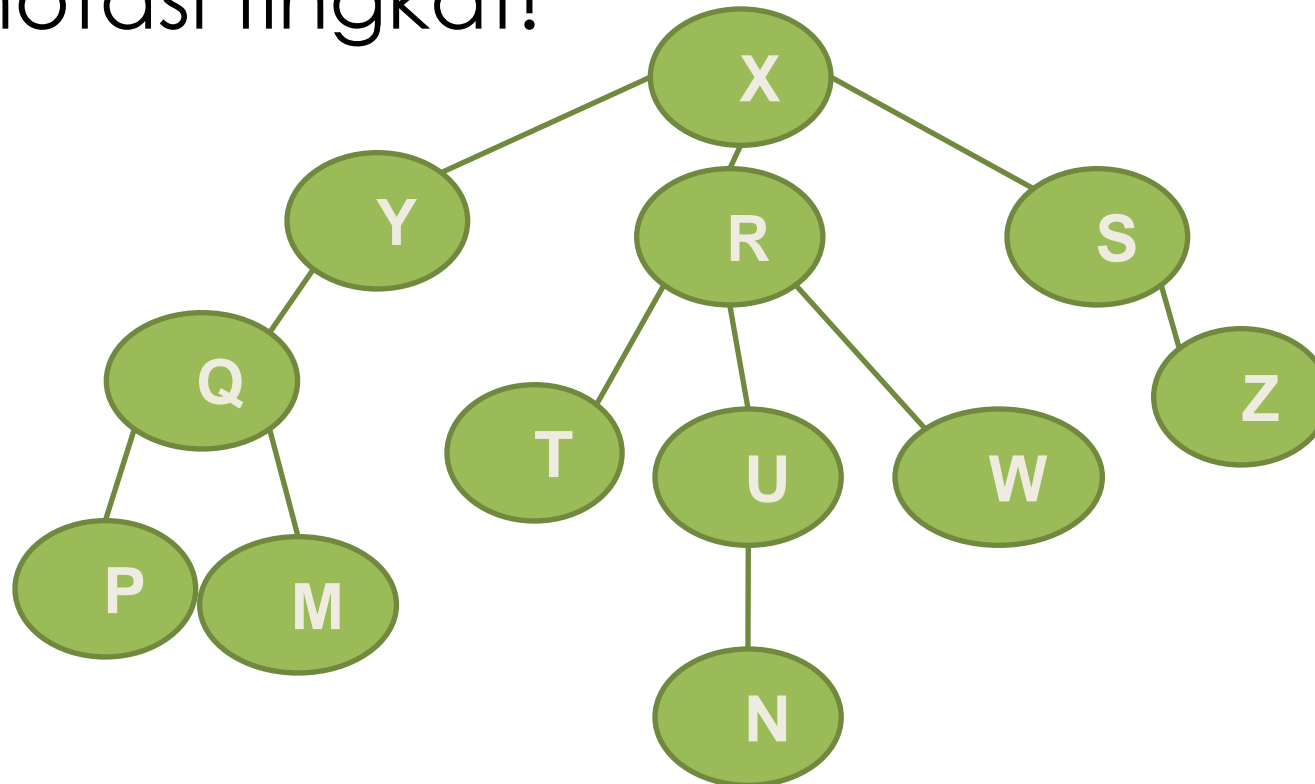
Tree Representation

- Notasi Kurung

$(A(B(D,E(I,J)),C(F,G,H)))$

Latihan

- Buat diagram venn, notasi kurung dan notasi tingkat!



• Identifikasikan !

- Ancestor (N) =
- Descendant (Y) =
- Parent (Z) =
- Child (Q) =
- Sibling (U) =
- Size =
- Height =
- Root =
- Leaf =
- Degree (R) =

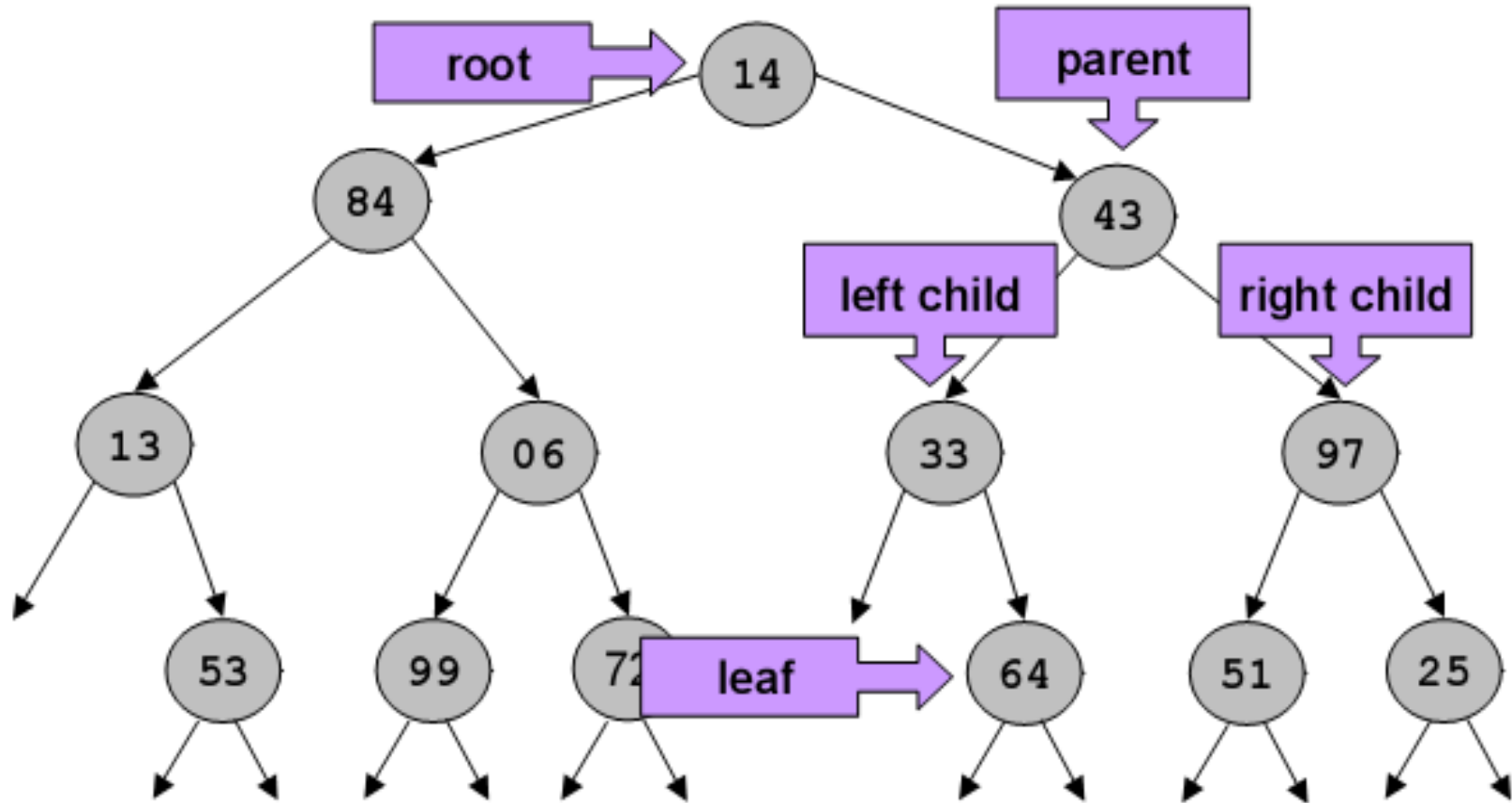
Binary Tree



Semakin tinggi pohon, semakin kuat angin meniupnya

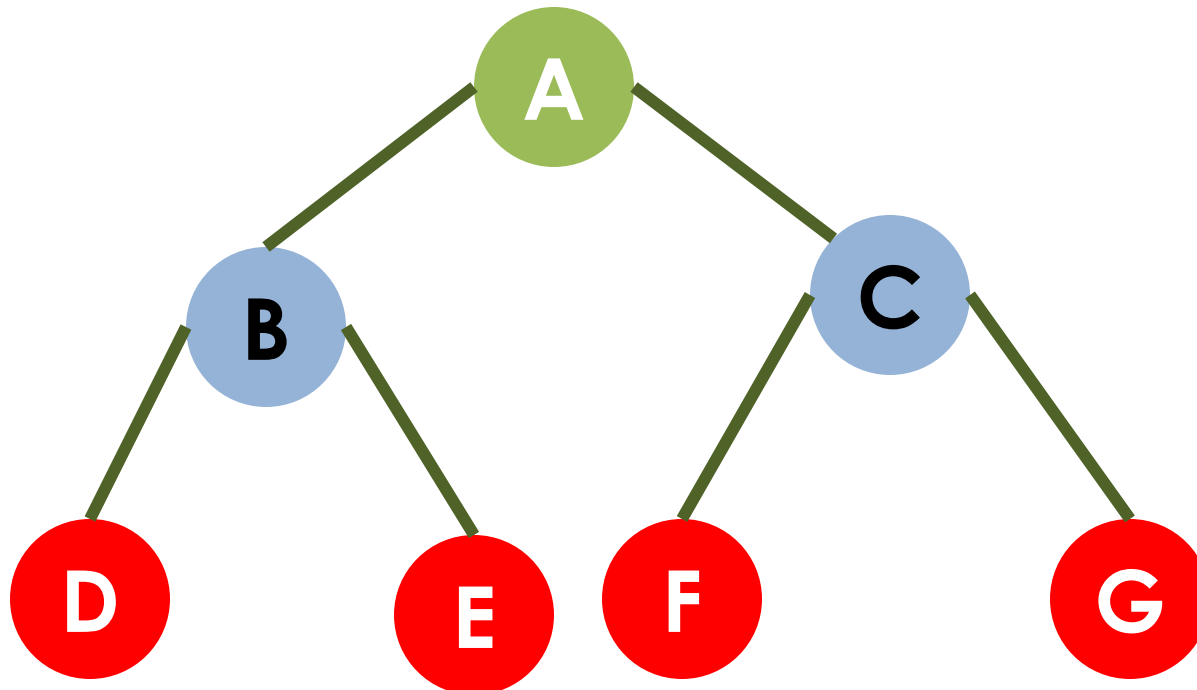
- Suatu tree dengan syarat bahwa tiap node hanya boleh memiliki maksimal **dua subtree** dan kedua subtree tersebut harus **terpisah**.
- Tiap node dalam binary tree hanya boleh memiliki **paling banyak** dua child.

Binary Tree



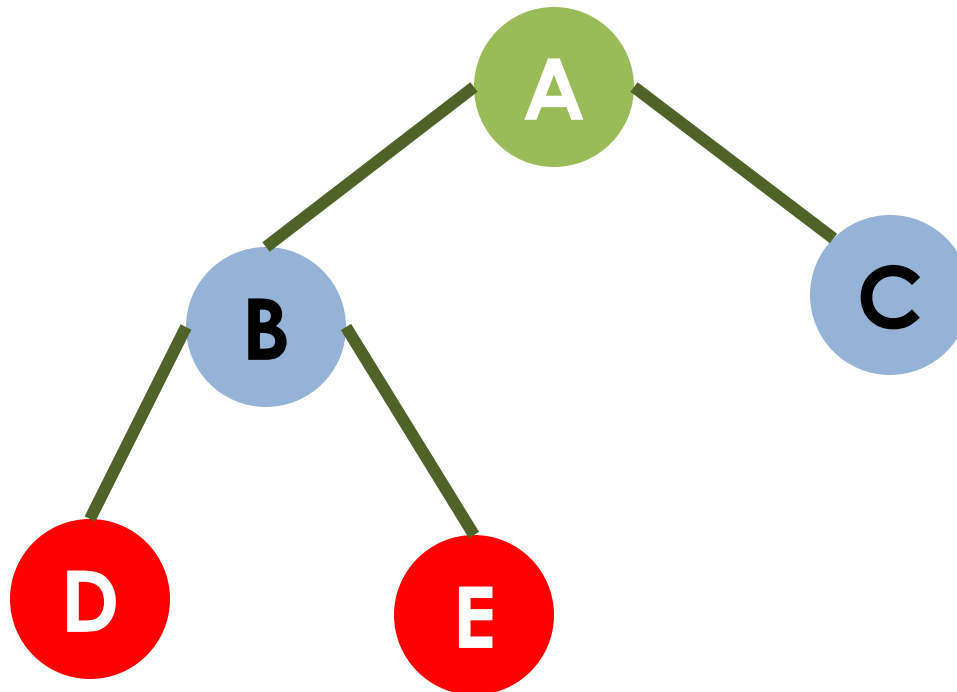
Binary Tree

- Full Binary Tree: semua node (kecuali leaf) pasti memiliki 2 anak dan tiap subtree memiliki panjang path yang sama.



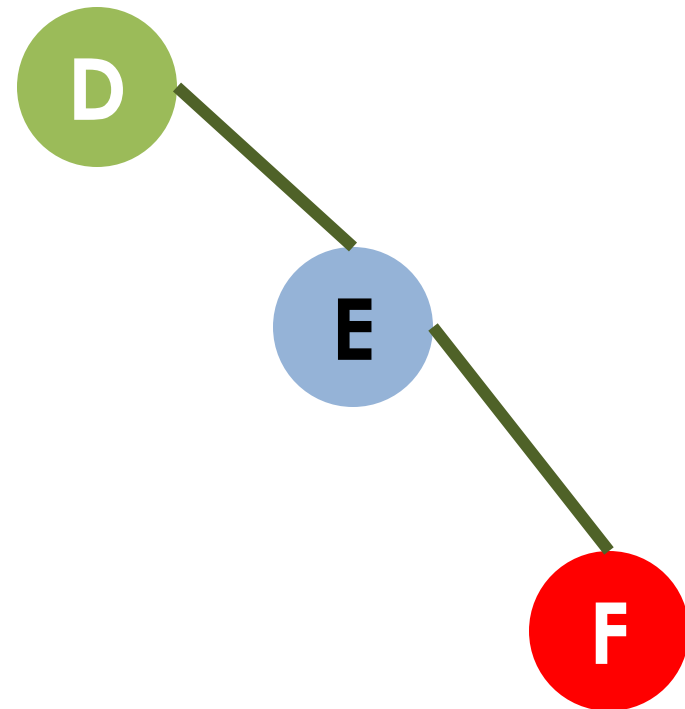
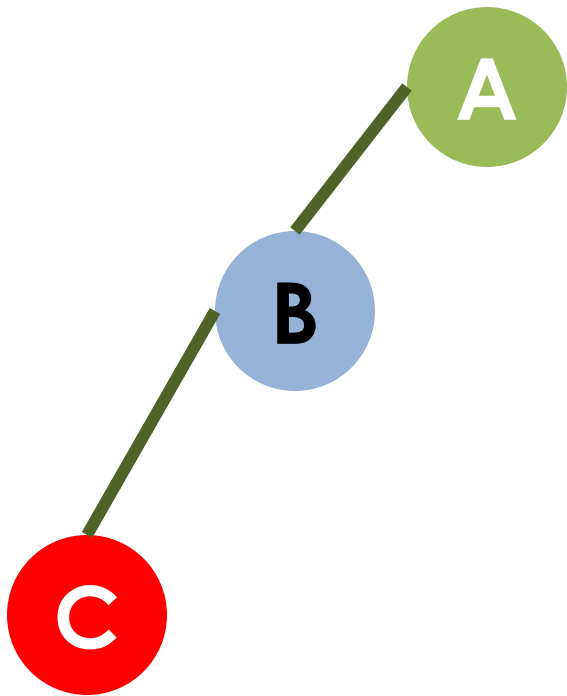
Binary Tree

- Complete Binary Tree: mirip dengan full binary tree, tapi tiap subtree boleh memiliki panjang path yang berbeda dan tiap node (kecuali leaf) memiliki 2 anak.



Binary Tree

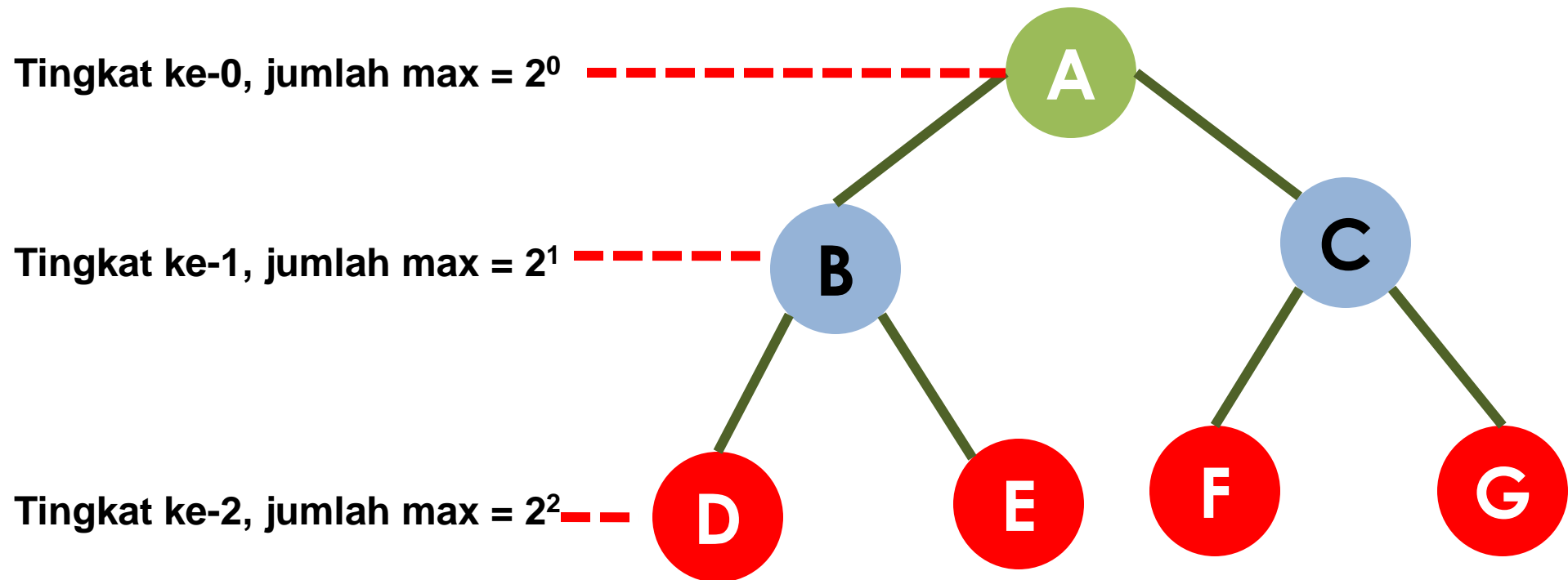
Skewed Binary Tree: binary tree yang semua nodenya (kecuali leaf) hanya memiliki satu anak.



Node pada Binary Tree

- Jumlah maksimum node pada setiap tingkat adalah 2^n , $n = \text{level/tingkat}$
- Node pada binary tree maksimum berjumlah $2^m - 1$, $m = \text{height}$

Node pada Binary Tree



Latihan

- Gambarkan pohon biner dengan ketentuan sbb :
 - Ancestor (M) = Z, X
 - Descendant (Y) = K, L
 - Parent (N) = Z
 - Child (Z) = M, N
 - Sibling (Y) = Z
 - Size = 7
 - Height = 3
 - Root = X
 - Leaf = K, L, M, N



Penelusuran Pohon Biner

Jika kau teriaki dan maki pohon, lambat laun pohon akan mati

Definisi

- Penelusuran seluruh node pada binary tree.
- Metode :

–Preorder

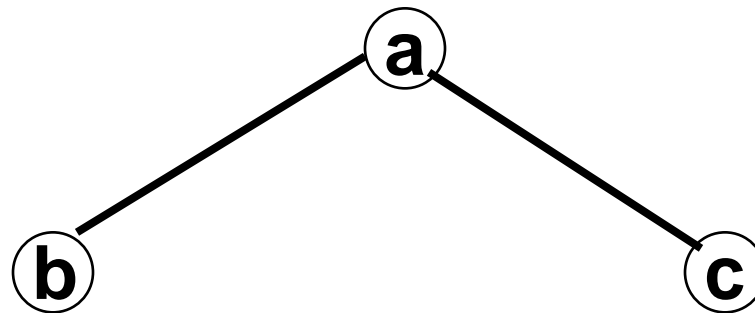
–Inorder

–Postorder

PreOrder Traversal

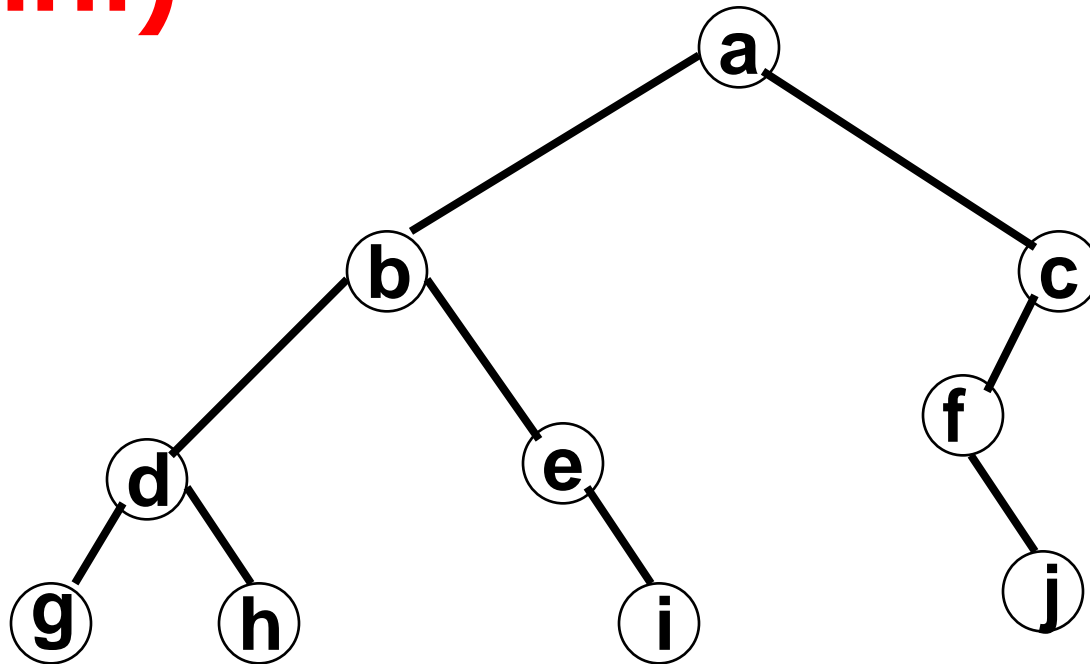
1. Cetak data pada root
2. Secara rekursif mencetak seluruh data pada subpohon kiri
3. Secara rekursif mencetak seluruh data pada subpohon kanan

Preorder Example (visit = print)



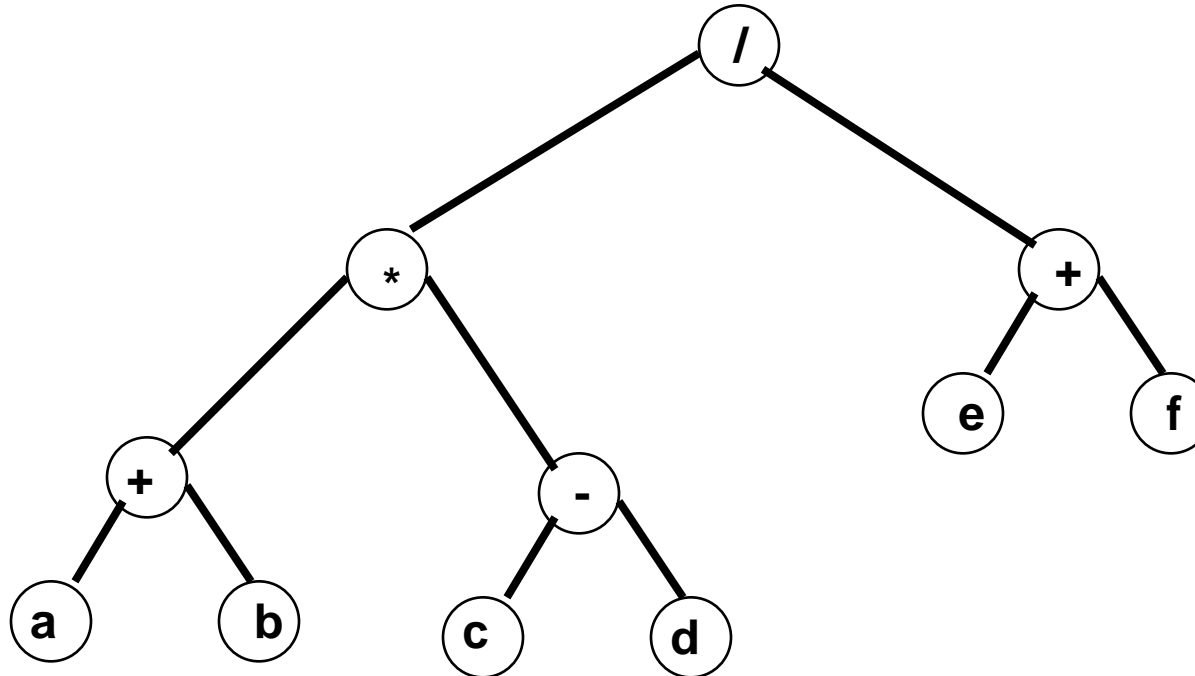
a b c

Preorder Example (visit = print)



a b d g h e i c f j

Preorder Of Expression Tree



/ * + a b - c d + e f

Gives prefix form of expression!

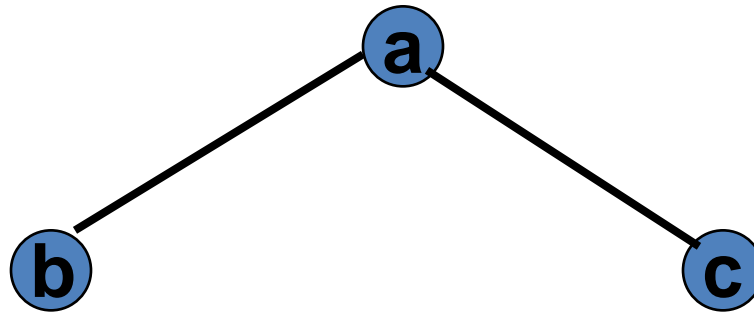
Script Preorder Of Expression Tree

```
void preorder(pohon ph)
{
    if (ph != NULL)
    {
        printf("%c ", ph->info);
        preorder(ph->kiri);
        preorder(ph->kanan);
    }
}
```

InOrder Traversal

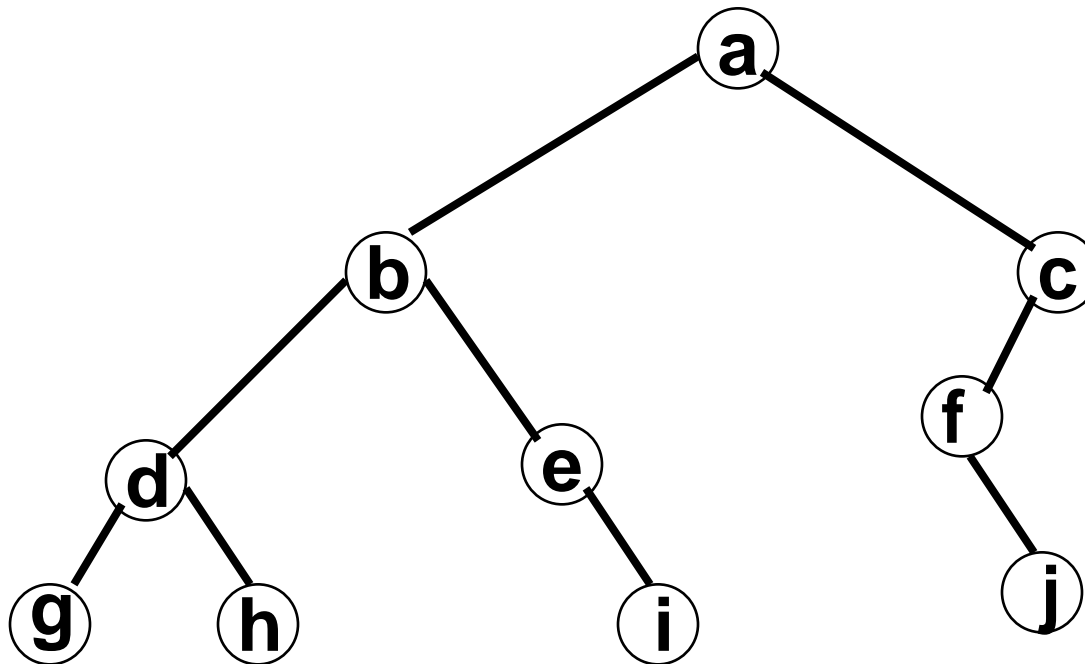
1. Secara rekursif mencetak seluruh data pada subpohon kiri
2. Cetak data pada root
3. Secara rekursif mencetak seluruh data pada subpohon kanan

Inorder Example (visit = print)



b a c

Inorder Example (visit = print)



g d h b e i a f j c

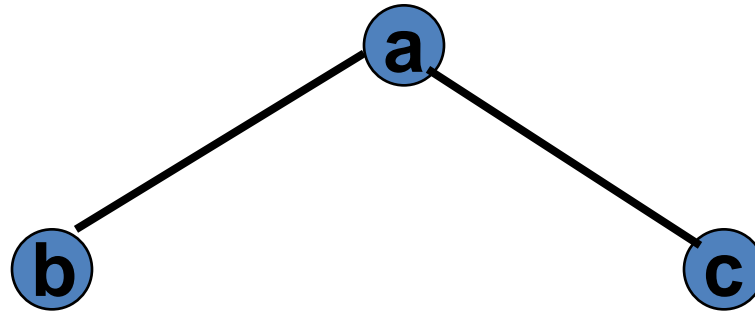
Script InOrder Of Expression Tree

```
void inorder(pohon ph)
{
    if (ph != NULL)
    {
        inorder(ph->kiri);
        printf("%c ", ph->info);
        inorder(ph->kanan);
    }
}
```

Postorder Traversal

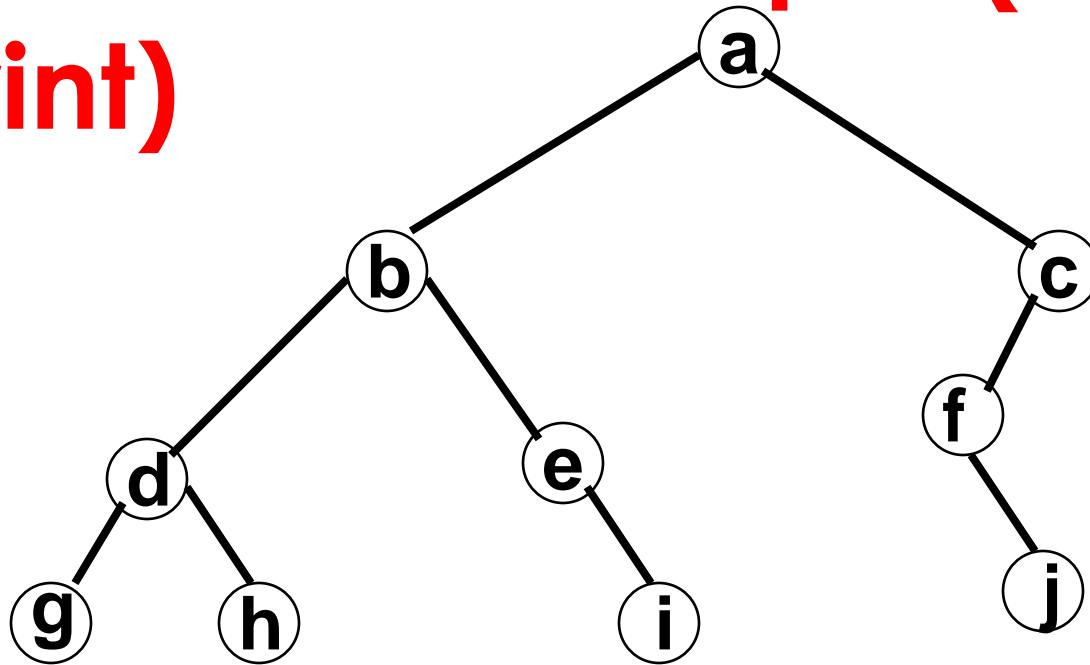
1. Secara rekursif mencetak seluruh data pada subpohon kiri
2. Secara rekursif mencetak seluruh data pada subpohon kanan
3. Cetak data pada root

Postorder Example (visit = print)



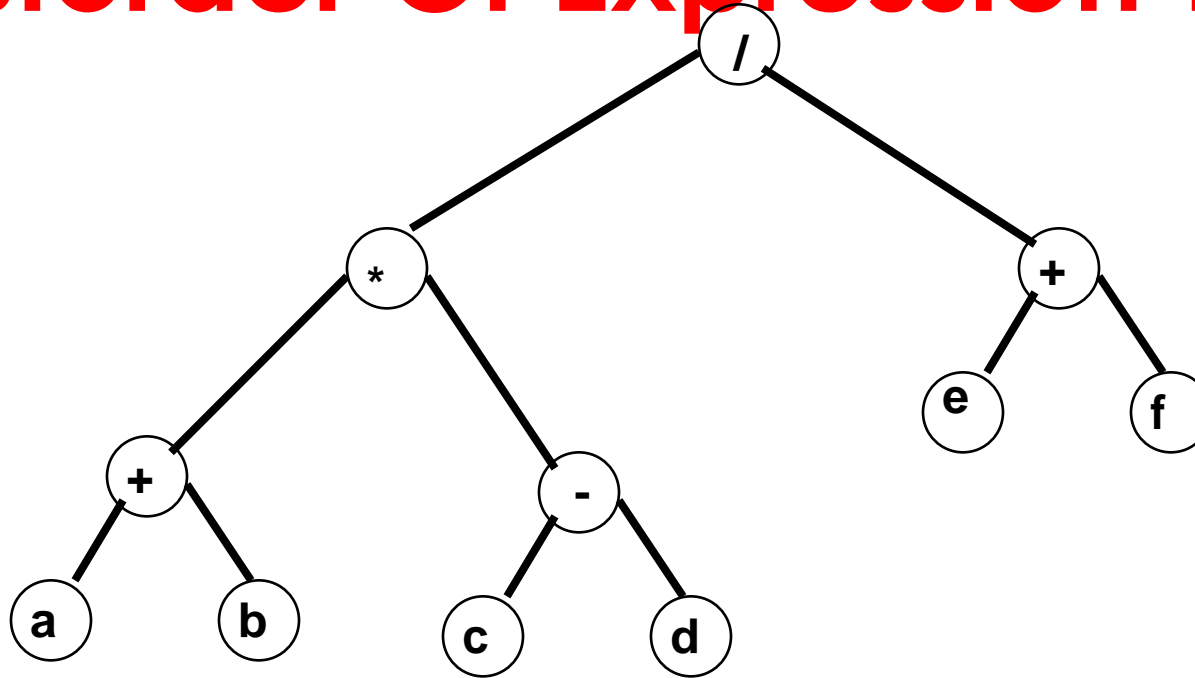
b c a

Postorder Example (visit = print)



g h d i e b j f c a

Postorder Of Expression Tree



a b + c d - * e f + /

Gives postfix form of expression!

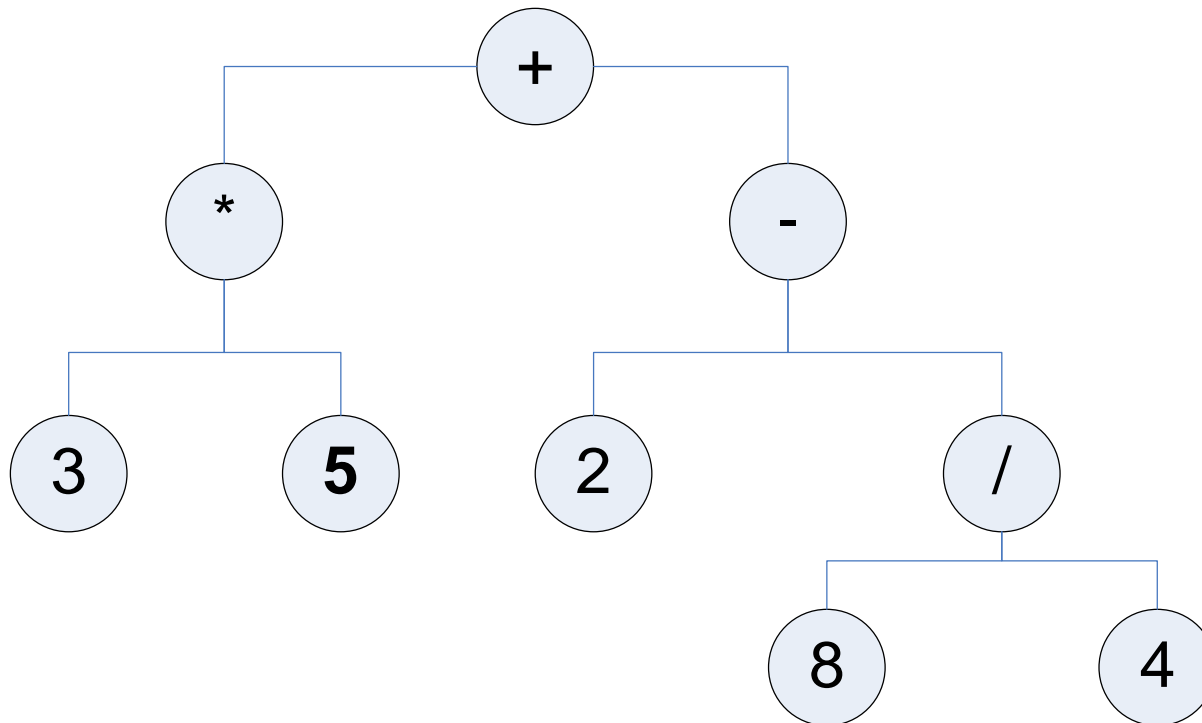
Script PostOrder Of Expression Tree

```
void postorder(pohon ph)
{
    if (ph != NULL)
    {
        postorder(ph->kiri);
        postorder(ph->kanan);
        printf("%c ", ph->info);
    }
}
```

Latihan

- Telusuri pohon biner berikut dengan menggunakan metode pre, in, post !

Latihan 1



Latihan 2

