



Institut Teknologi
Telkom
Purwokerto

Teknik Informatika - Fakultas Informatika

Pertemuan 12 – Tree

Author: **Wahyu Andi Saputra [WAA]**

Outline

Konsep Tree

Tree pada Struktur Data

Binary Tree

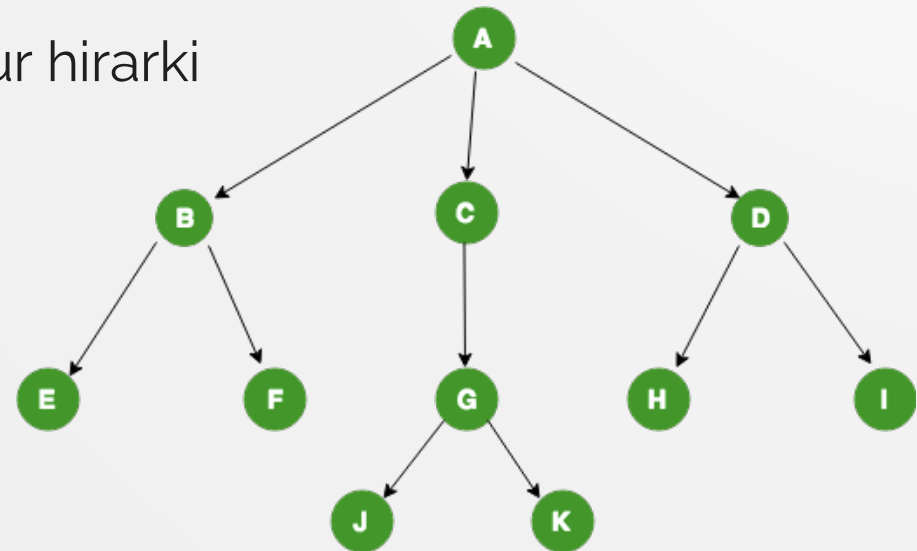
Tree Traversal



Konsep Tree

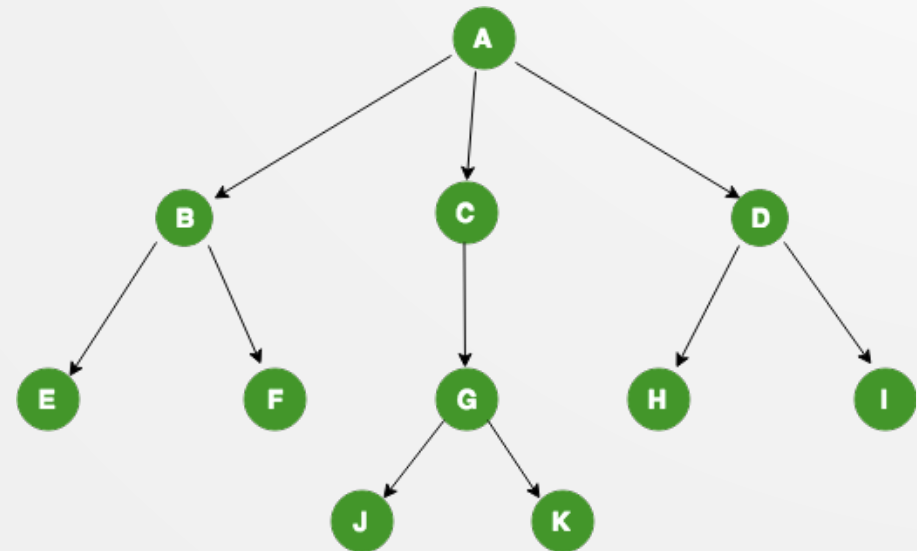
Konsep Tree

- adalah kumpulan node yang saling terhubung satu sama lain dalam suatu kesatuan yang membentuk layaknya struktur sebuah pohon.
- Merepresentasikan suatu struktur hirarki



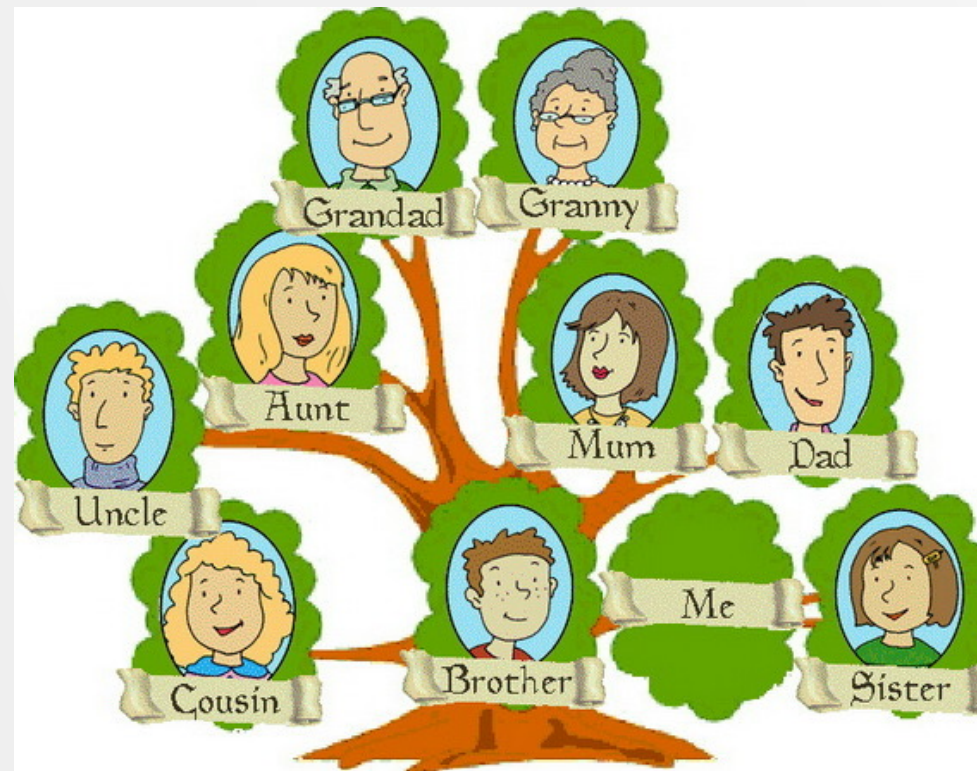
Konsep Tree

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



Implementasi Tree

- Pohon keluarga



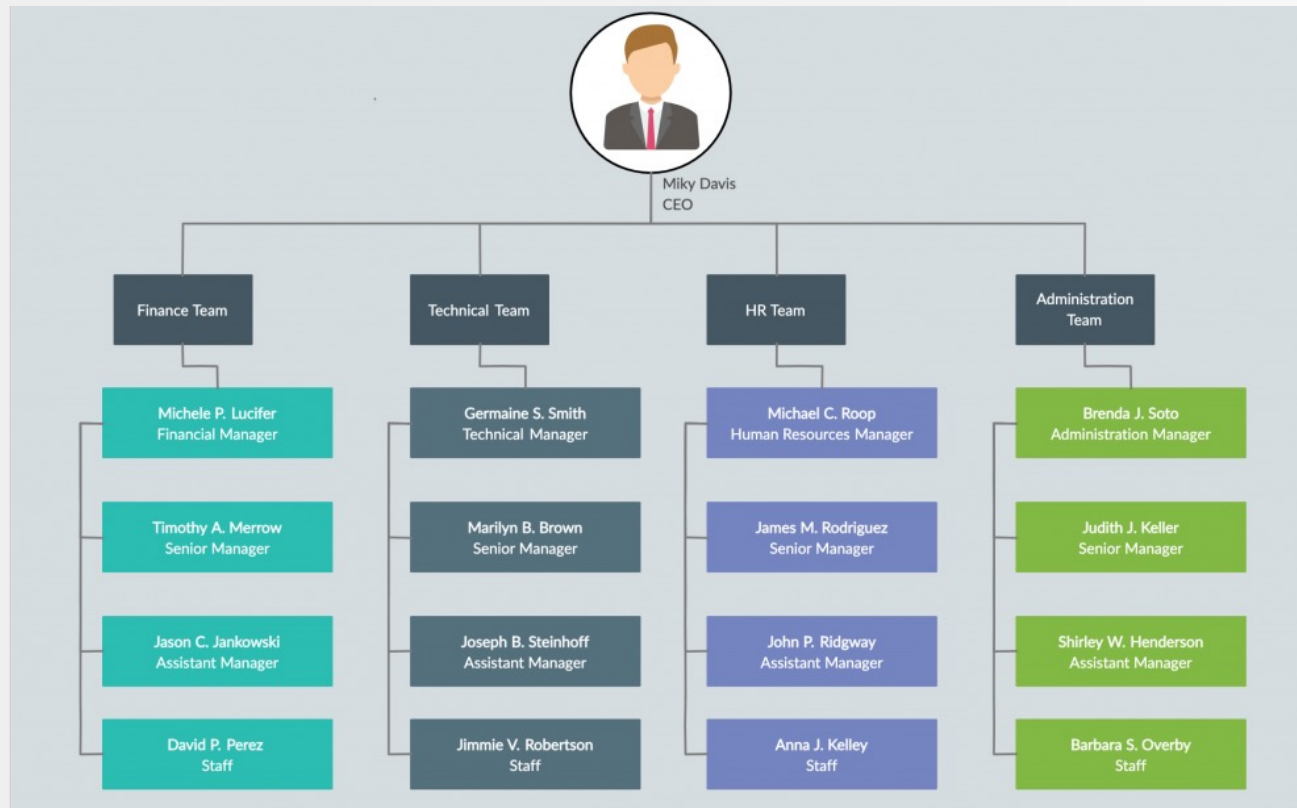
Implementasi Tree

- Penjadwalan turnamen



Implementasi Stack

- Organization Structure





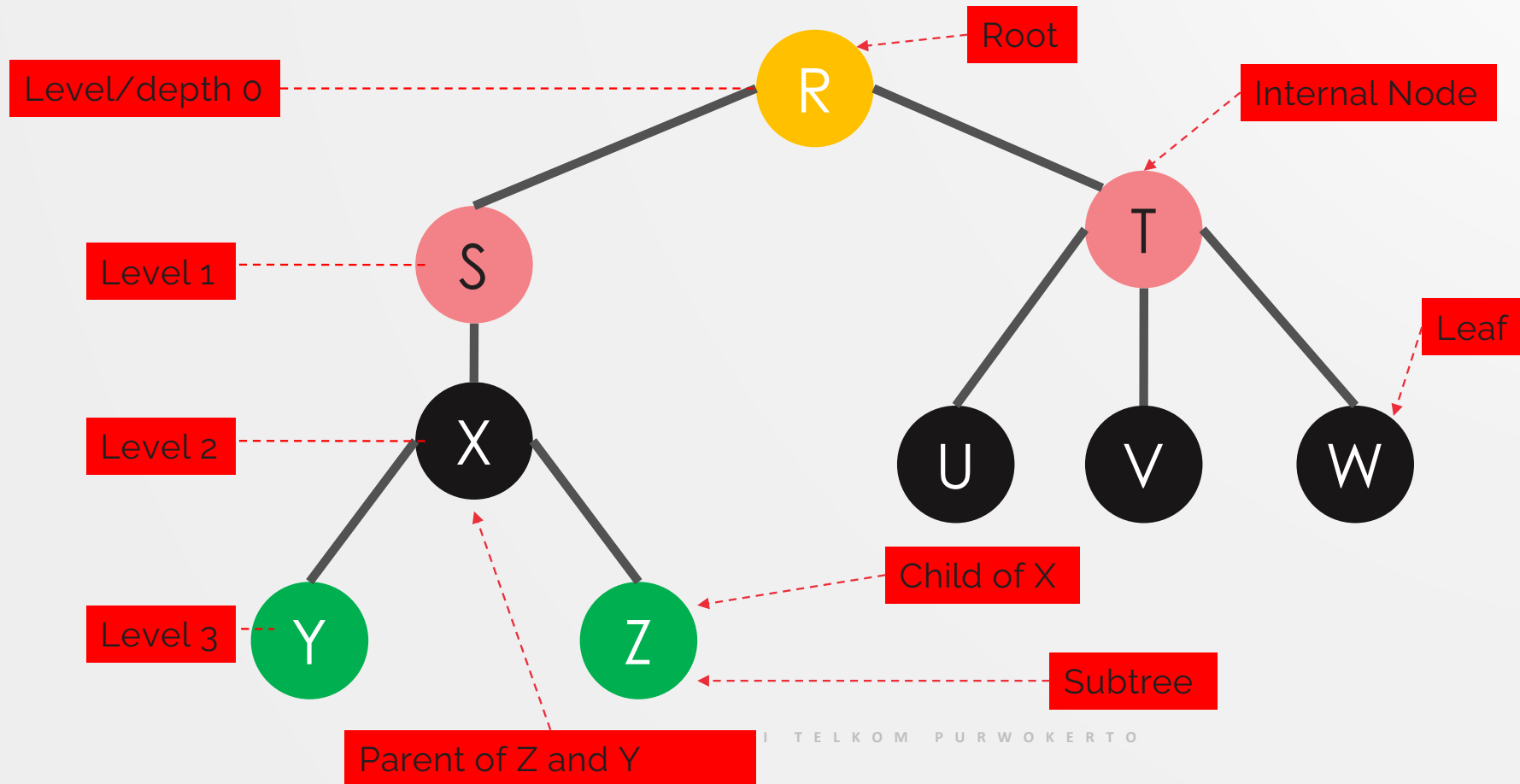
Implementasi Stack

- Pengelolaan memori pada system operasi
- Operasi aritmetika pada cara kerja kalkulator
- Penyelesaian pencarian rute labirin



Tree pada Struktur Data

Anatomi Tree

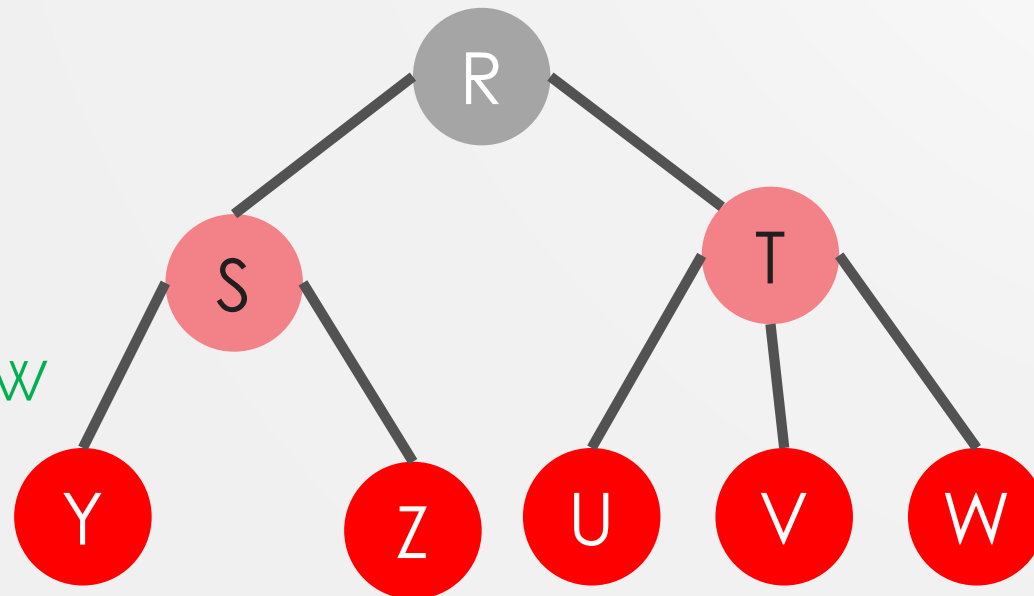


Anatomi Tree

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

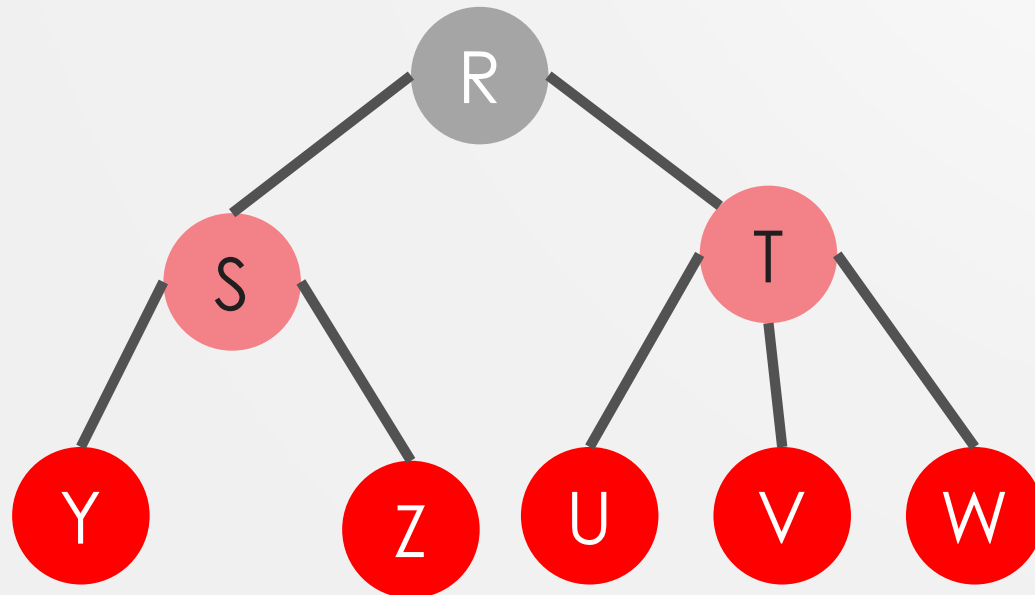
Anatomi Tree

- Predecessor (Y,S,Z,T) = R
- Successor (T) = U, V, W
- Ancestor (U) = T, R
- Descendant (T) = U, V, W
- Parent (Y) = S
- Child (R) = S, T



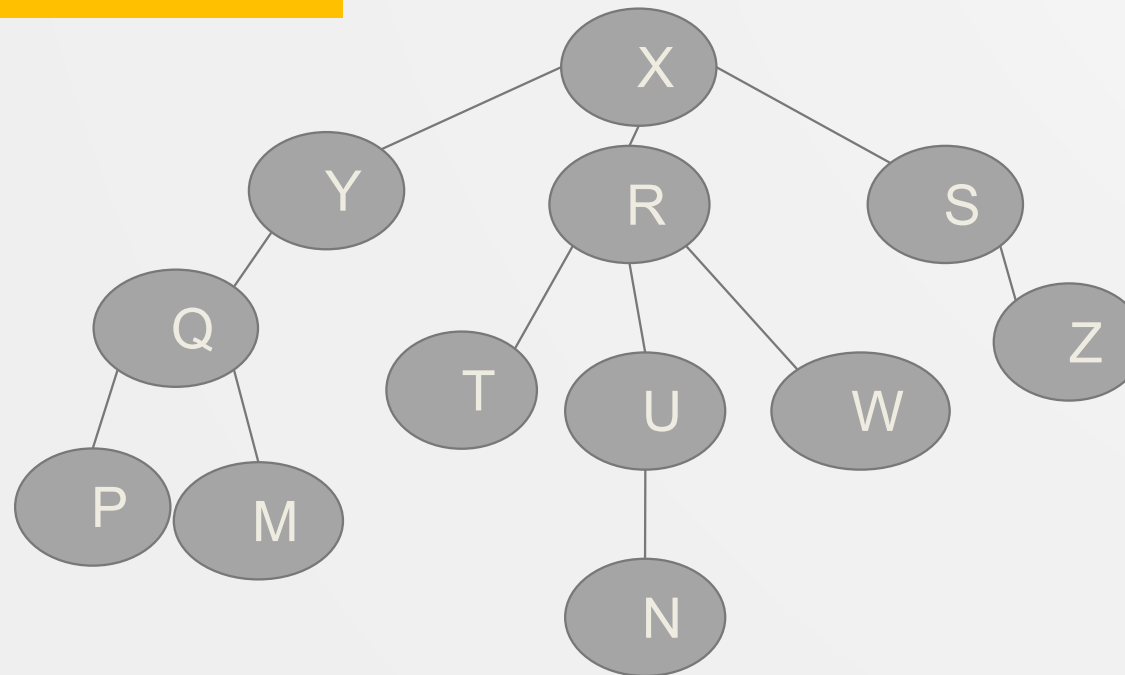
Anatomi Tree

- Sibling (U) = V, W
- Size = 8
- Height = 3
- Root = R
- Leaf = Y, Z, U, V, W
- Degree (T) = 3



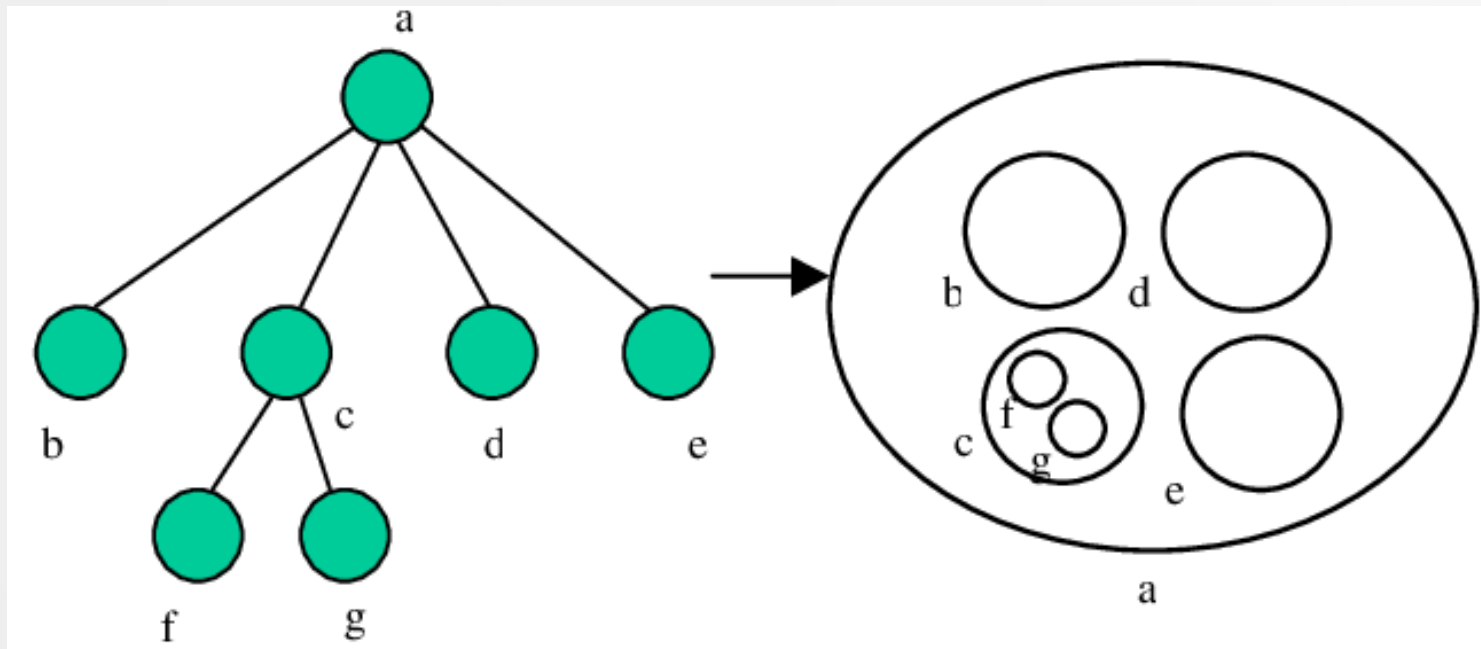
Anatomi Tree

Jelaskan anatomi tree
dari tree berikut!



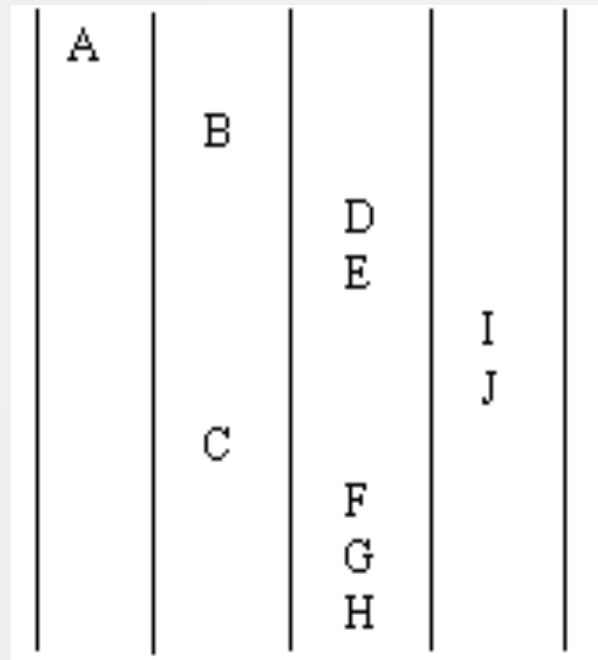
Representasi Tree

- Diagram Venn



Representasi Tree

- Notasi Tingkat



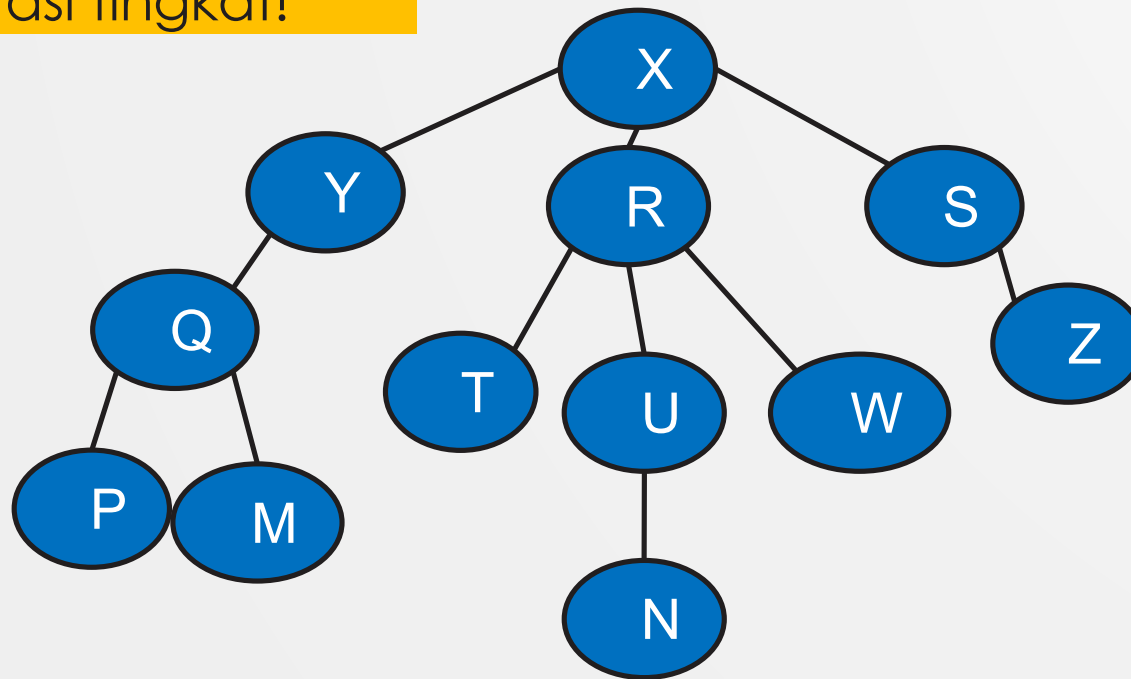
Representasi Tree

- Notasi Kurung

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

Representasi Tree

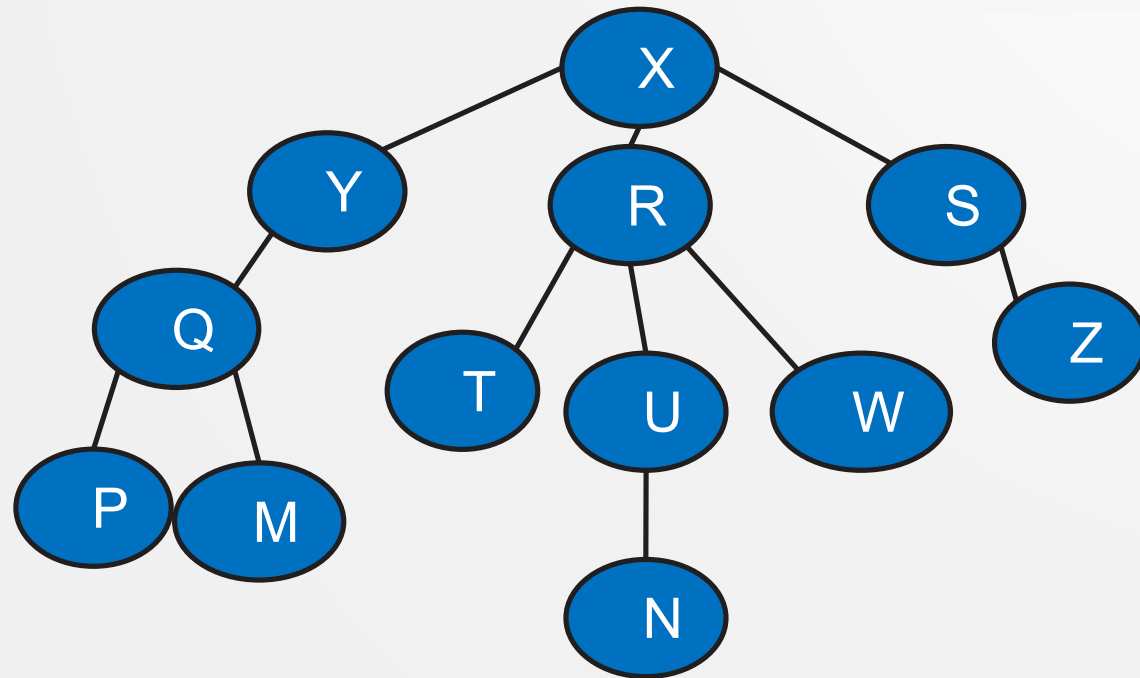
Buat diagram venn, notasi kurung dan notasi tingkat!



Representasi Tree

Identifikasikan!

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



Representasi Tree

Gambarkan pohon binary-nya!

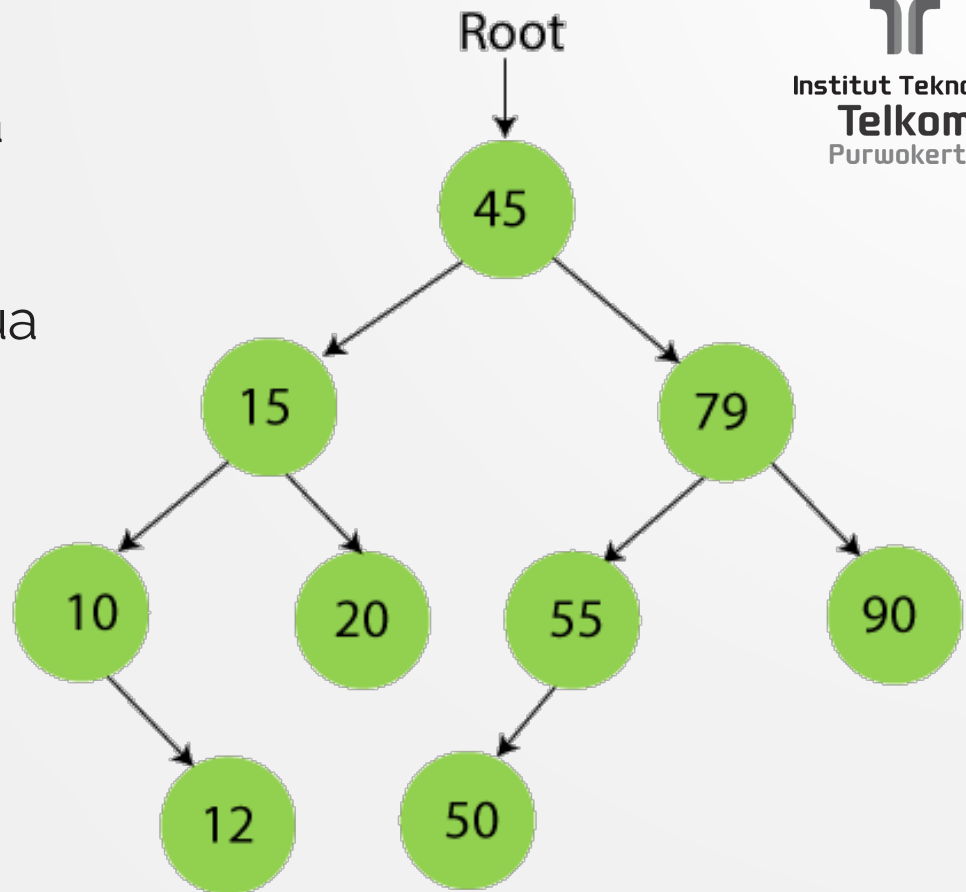
- Ancestor (M) = Z, X
- Descendant (Y) = K, L
- Parent (N) = Z
- Child (Z) = M, N
- Sibling (Y) = Z
- Size = 7
- Height = 3
- Root = X
- Leaf = K, L, M, N



Binary Tree

Binary Tree

- 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.

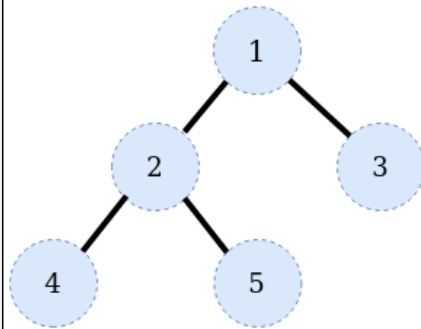


Kategori Binary Tree

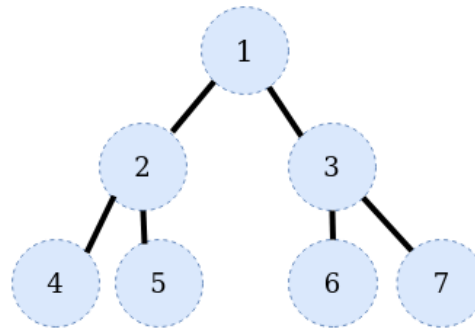


Institut Teknologi
Telkom
Purwokerto

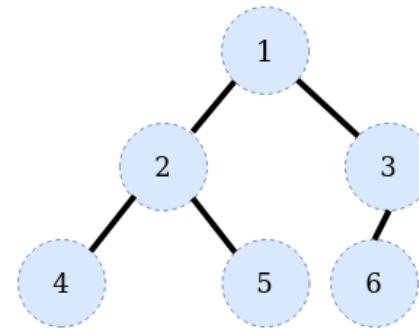
Full Binary Tree



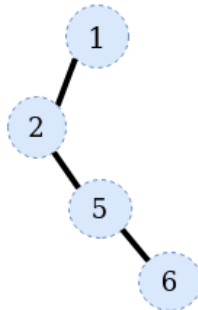
Perfect Binary Tree



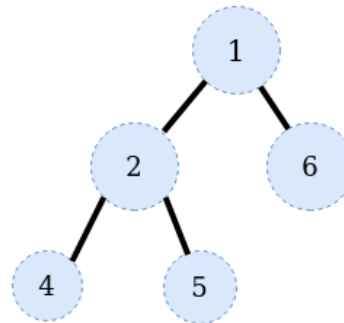
Complete Binary Tree



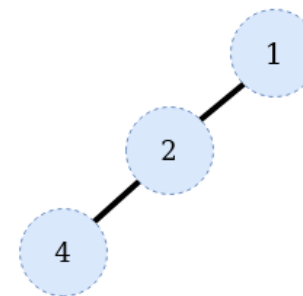
Degenerate Tree



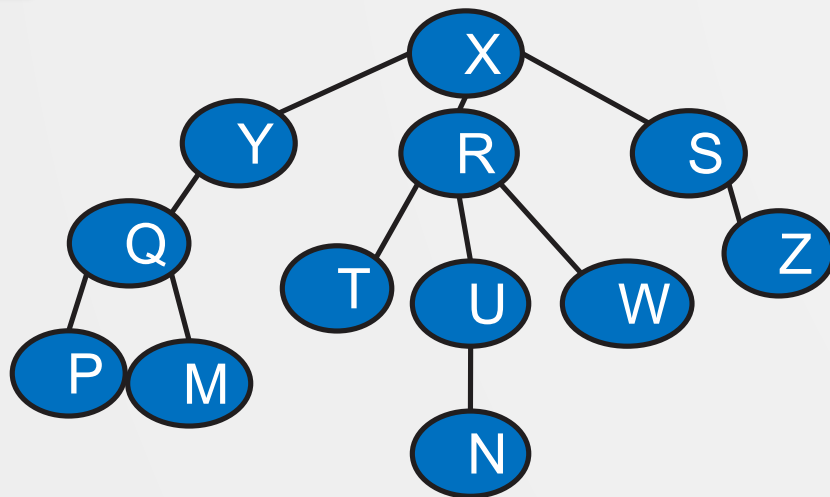
Balanced Tree



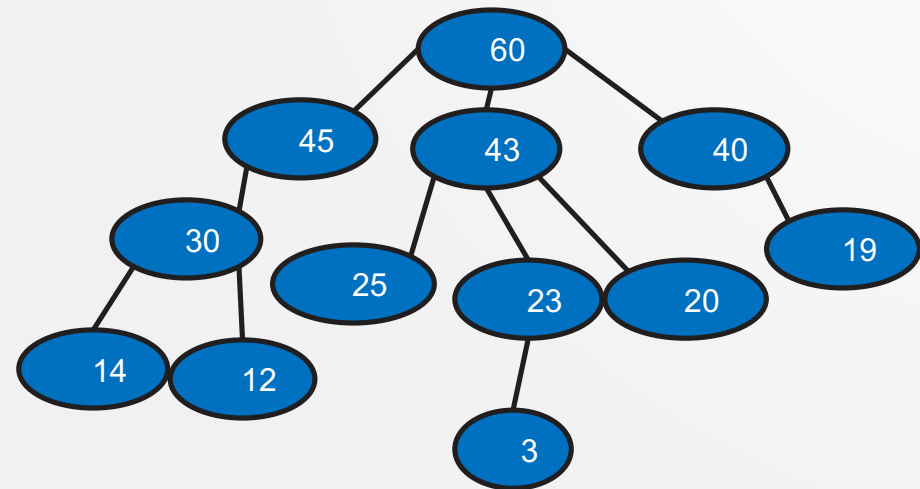
Skewed Tree -Left



Kategori Binary Tree



■ Unordered tree



■ Ordered tree

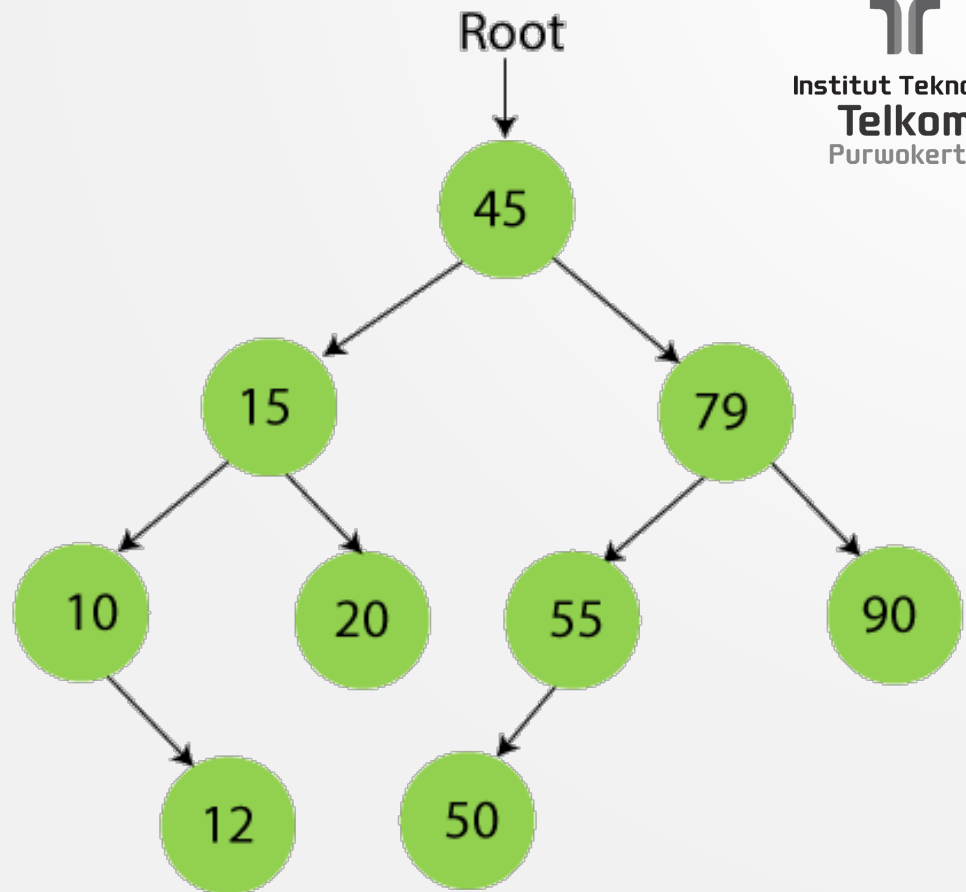
Binary Tree

- Node pada suatu binary tree maksimum berjumlah

$$2^m - 1, m = \text{height}$$

- Jumlah maksimum node pada setiap tingkat adalah

$$2^n, n = \text{level/tingkat}$$



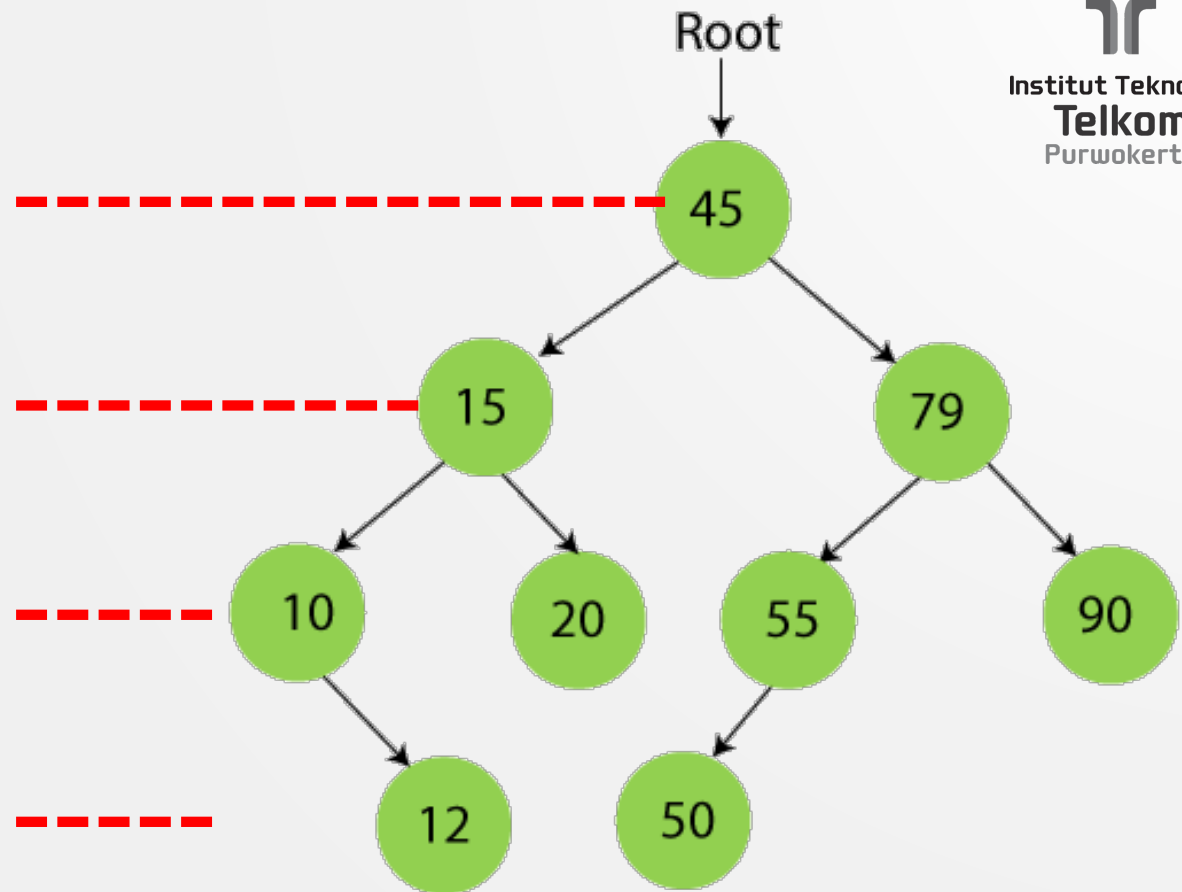
Binary Tree

Level-0, jumlah max = 2^0

Level-1, jumlah max = 2^1

Level-2, jumlah max = 2^2

Level-3, jumlah max = 2^3





Tree Traversal

Tree Traversal

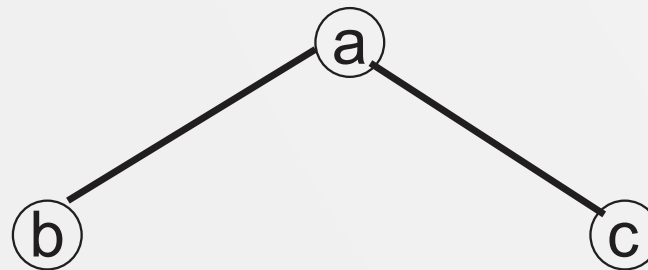
- Penelusuran seluruh node pada binary tree.
- Metode :

- **Preorder**
- **Inorder**
- **Postorder**

Tree Traversal: PreOrder

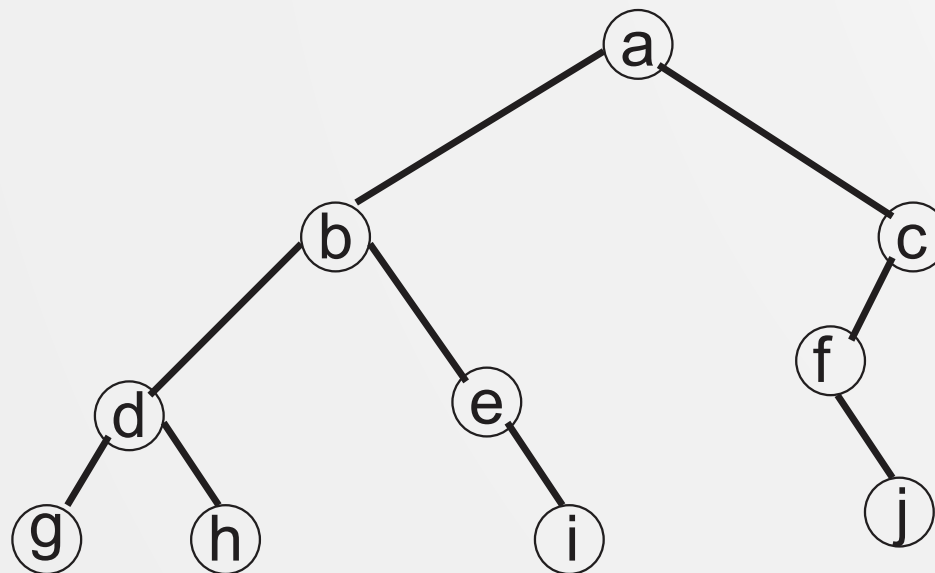
Secara rekursif di setiap node:

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



a b c

Tree Traversal: PreOrder

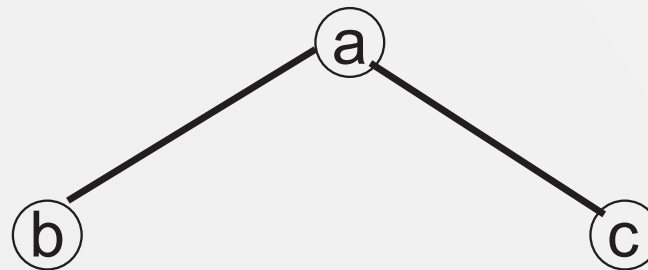


a b d g h e i c f j

Tree Traversal: InOrder

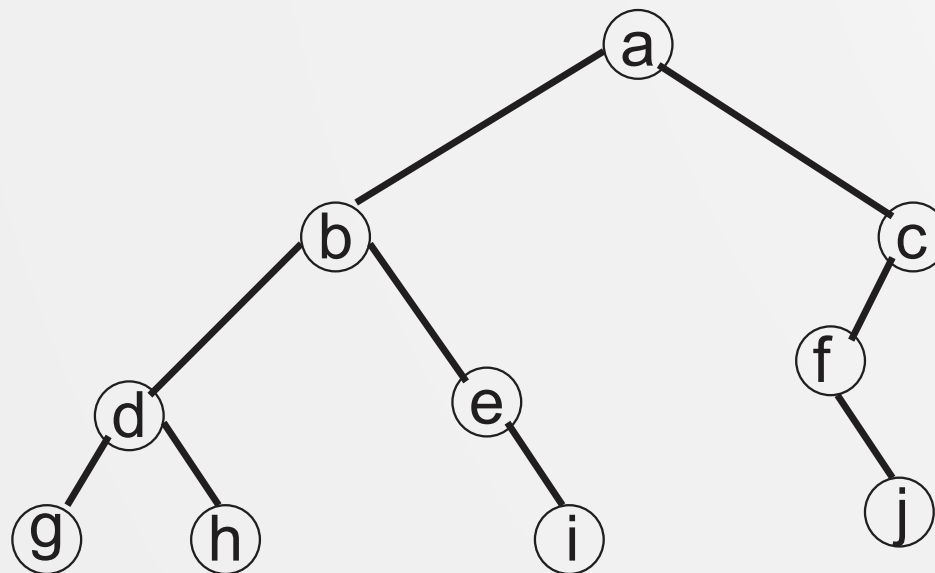
Secara rekursif di setiap node:

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



b a c

Tree Traversal: InOrder

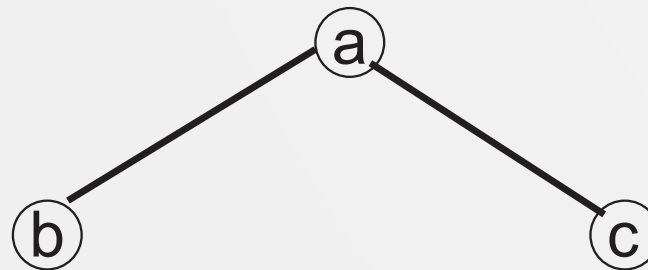


g d h b e i a f j c

Tree Traversal: **PostOrder**

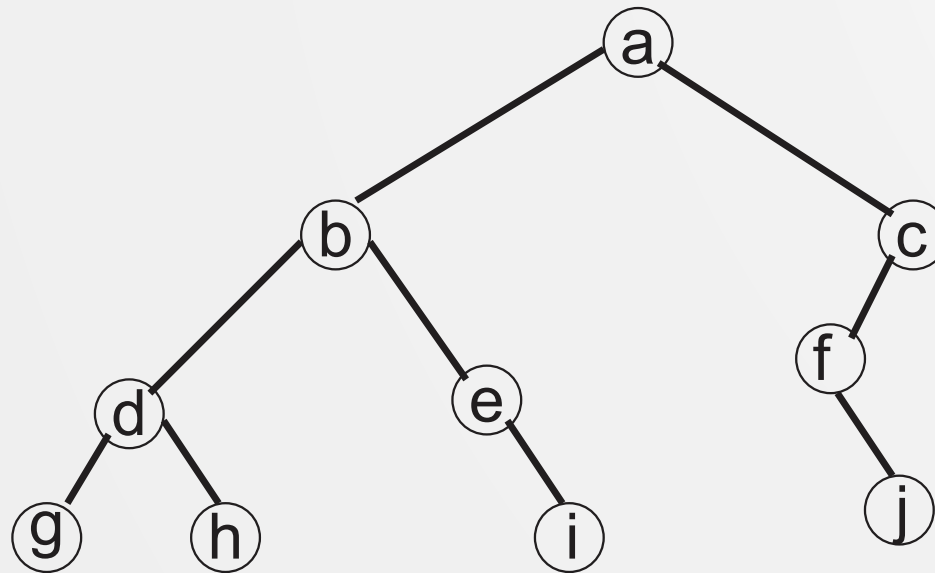
Secara rekursif di setiap node:

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



b c a

Tree Traversal: **PostOrder**



g d h i e b j f c a



Institut Teknologi
Telkom
Purwokerto

TERIMA KASIH