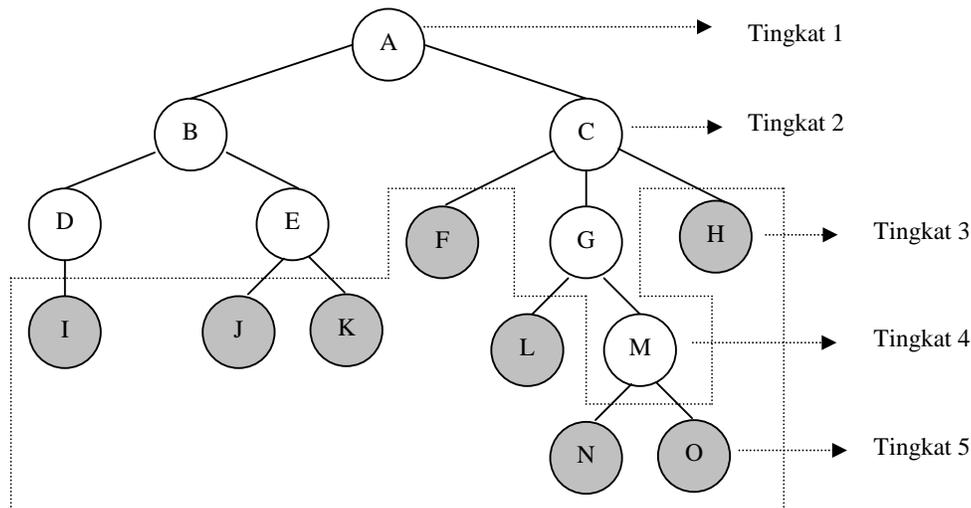


BAB IV POHON

Definisi Pohon

“Struktur pohon merupakan kumpulan elemen yang salah satu elemennya disebut akar dan sisa elemennya terpecah menjadi sejumlah himpunan yang saling tidak berhubungan satu sama lainnya yang disebut cabang”.

- **Simpul /node /vertex** adalah elemen pohon yang berisi informasi/data dan penunjuk percabangan. Pada gambar 1 pohon dengan 15 simpul.
- **Tingkat (level)** suatu simpul ditentukan dengan pertama kali menentukan akar sebagai bertingkat 1. Jika suatu simpul dinyatakan sebagai tingkat N, maka simpul-simpul yang merupakan anaknya dikatakan berada dalam tingkat N + 1. Ada pula yang menyatakan bahwa akar berada pada tingkat 0 dan simpul-simpul lainnya dinyatakan bertingkat 1 lebih tinggi .

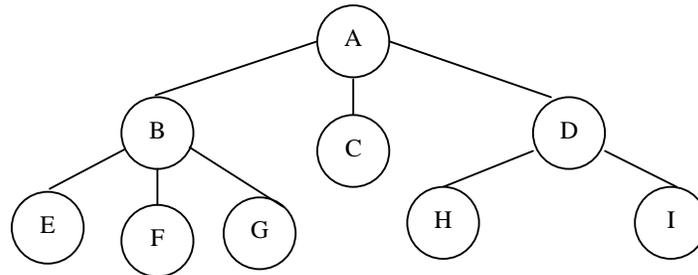


Gambar 1

- **Derajat (degree) suatu simpul** dinyatakan sebagai banyaknya generasi atau turunan dari simpul tersebut. Pada gambar 1 simpul A mempunyai derajat 2, B mempunyai derajat 2, C mempunyai derajat 3. **Simpul berderajat 0 disebut dengan daun (leaf)**. Simpul-simpul F, H, I, J, K, L, N, O berderajat 0, disebut dengan daun. Daun juga sering disebut dengan simpul luar (external node) , sehingga simpul lain kecuali akar juga sering disebut dengan simpul dalam (internal node).
- **Tinggi (height) atau kedalaman (depth)** dari suatu pohon adalah tingkat maksimum dari simpul dalam pohon tersebut dikurangi dengan 1. Pohon gambar 1 mempunyai tinggi atau kedalaman 4.
- **Ancesor** suatu simpul adalah semua simpul yang terletak dalam satu jalur dengan simpul tersebut dari akar sampai simpul yang ditinjau. Contoh ancestor dari simpul L adalah A, C, G.

- Predecessor adalah simpul yang berada di atas simpul yang ditinjau. Contoh : Predecessor D adalah B.
- Successor adalah simpul yang berada di bawah simpul yang ditinjau. Contoh : Successor D adalah I.
- Descendant adalah seluruh simpul yang terletak sesudah simpul tertentu dan terletak pada jalur yang sama. Contoh : Descendant E adalah J dan K.
- Sibling adalah simpul-simpul yang memiliki parent yang sama dengan simpul yang ditinjau. Contoh : Sibling J adalah K
- Parent adalah simpul yang berada satu level di atas simpul yang ditinjau. Contoh : Parent J adalah E
J adalah E
- **Hutan (forest)** adalah kumpulan sejumlah pohon yang tidak saling berhubungan. Pada gambar 1 jika simpul A dihapus maka akan diperoleh sebuah hutan.

Struktur data pada bentuk struktur pohon dalam hubungan logiknya atau relasi antara simpul satu dengan simpul lainnya dianalogikan dengan relasi dalam sebuah keluarga atau dalam sebuah garis keturunan.



B,C dan D bersaudara. **E,F dan G** bersaudara. **H dan I** bersaudara. **A** merupakan orang tua (parent) dari **B,C dan D**. **B** merupakan orang tua dari **E,F dan G**. **D** merupakan orang tua dari **H dan I**.

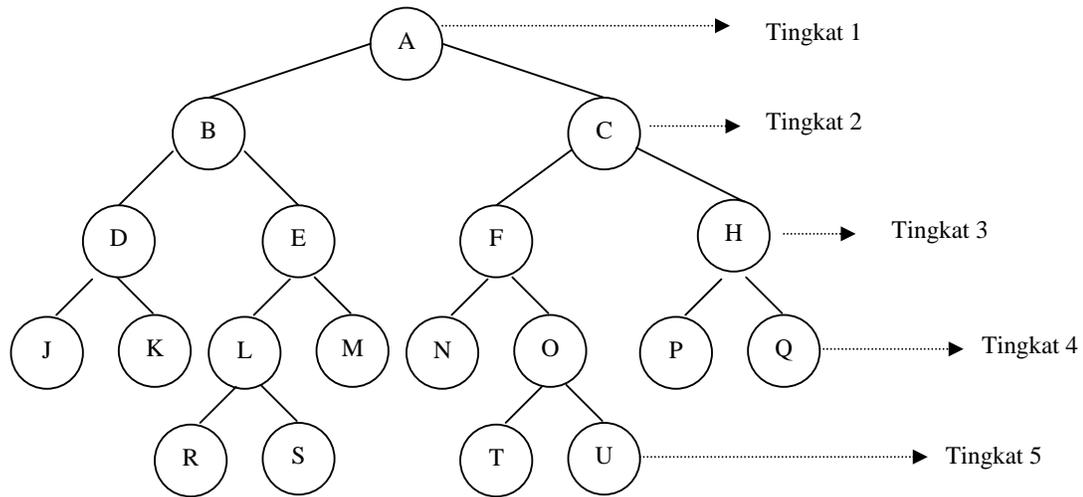
Pohon Biner

Pohon biner (binary tree) bisa didefinisikan sebagai suatu kumpulan simpul yang mungkin kosong atau mempunyai akar dan subpohon yang saling terpisah yang disebut dengan subpohon kiri (left subtree) dan sub pohon kanan (right tree)

Karakteristik pohon biner yaitu,

- Setiap simpul paling banyak hanya mempunyai dua buah anak.
- Derajat tertinggi dari simpul dalam pohon biner adalah dua.
- Banyaknya simpul maksimum pada tingkat $N = 2^{n-1}$. Pohon biner lengkap tingkat 4 mempunyai daun = 8 dan banyaknya simpul yang bukan daun termasuk akar =7.
- Dalam pohon biner dimungkinkan tidak mempunyai simpul.

Pohon biner lengkap (complete binary tree) bertingkat N, adalah sembarang pohon biner yang semua daunnya terdapat pada tingkat N dan semua simpul yang bertingkat lebih kecil dari N selalu mempunyai cabang kiri dan kanan. Gambar dibawah ini merupakan pohon biner tingkat 4 tetapi bukan pohon biner lengkap tingkat 5.



Sebuah pohon yang bukan pohon biner bisa diubah menjadi sebuah pohon biner , caranya :

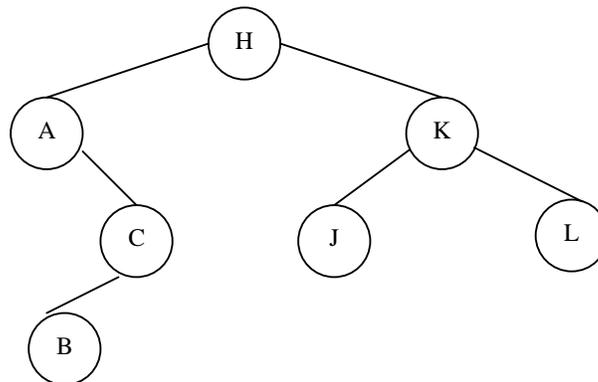
- Setiap simpul pohon biner mengandung field-field informasi, pada cabang kiri dan kanan.
- Cabang kiri merupakan anak paling kiri dari simpul pada pohon yang bukan biner.
- Sedangkan cabang kanannya merupakan saudara dari simpul pada pohon yang bukan biner.

Membuat Pohon Biner

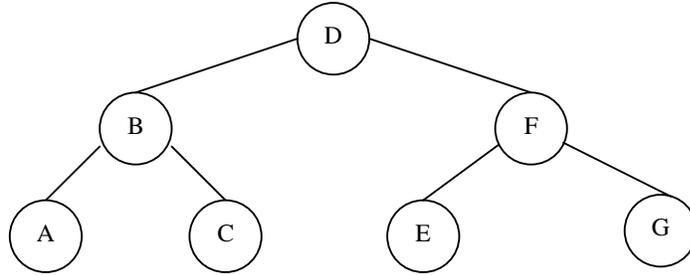
Algoritma : Simpul yang berisi informasi yang nilainya “lebih besar” dari simpul parent akan diletakkan sebagai cabang kanan, jika “lebih kecil” akan ditempatkan sebagai cabang kiri. **KIRI < KANAN.**

Contoh : Simpan data **H A K C B J L**

Karakter pertama ‘**H**’ dilatakkan sebagai akar. Karakter ‘**K**’ karena lebih besar dari ‘**H**’ akan menempati cabang kanan. Karakter ‘**A**’ karena lebih kecil dari ‘**H**’ akan menempati cabang kiri dari ‘**H**’. Kemudian, karena karakter ‘**C**’ lebih kecil dari ‘**H**’ dan labih besar dari ‘**A**’, maka ia diletakkan sebagai cabang kanan dari ‘**A**’. Proses ini berlangsung sama sampai semua masukan diproses.



Simpan data **D B F A C E G**



Traversal Pohon Biner

Traversal pohon biner adalah :

Proses mendatangi setiap simpul dari pohon secara sistematis masing-masing satu kali, dimana pada setiap simpul yang dikunjungi dilakukan suatu proses pengolahan data

Ada 3 cara traversal :

- a. Traversal postorder (LRS)

Proses pertama kali bergerak ke arah subpohon bagian kiri, selanjutnya subpohon sebelah kanan dan terakhir simpul (simpul root)

- b. Traversal inorder (LSR)

Proses dimulai dari subpohon sebelah kiri selanjutnya simpul root dan selanjutnya subpohon sebelah kanan.

- c. Traversal preorder (SLR)

Proses dimulai dari simpul (root) ke arah subpohon kiri, selanjutnya bergerak ke subpohon kanan (SLR).

S = proses mengunjungi simpul.

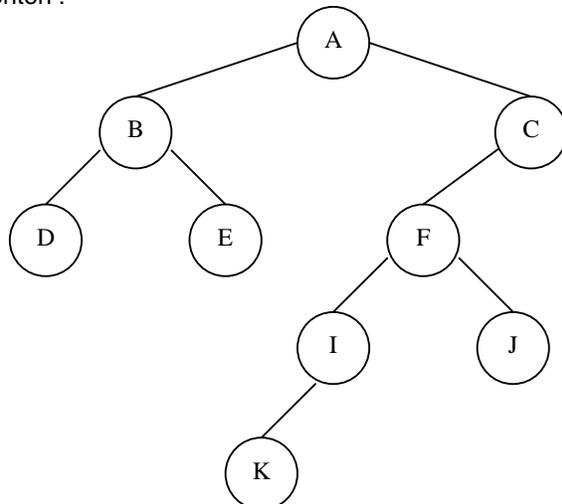
L = bergerak ke arah subpohon bagian kiri.

R = bergerak ke arah subpohon bagian kanan.

Traversal lain (traversal level order)

Kunjungan simpul dimulai dari simpul bertingkat satu, hingga simpul bertingkat n.

Contoh :



Preorder : A B D E C F I K J
Postorder : D E B K I J F C A
Inorder : D B E A K I F J C
Level order : A B C D E F I J K

Perbedaan proses

Preorder : proses dilakukan sebelum penelusuran dua bagian pohon.

Inorder : proses dilakukan diantara penelusuran dua bagian pohon.

Postorder : proses dilakukan setelah penelusuran dua bagian pohon.

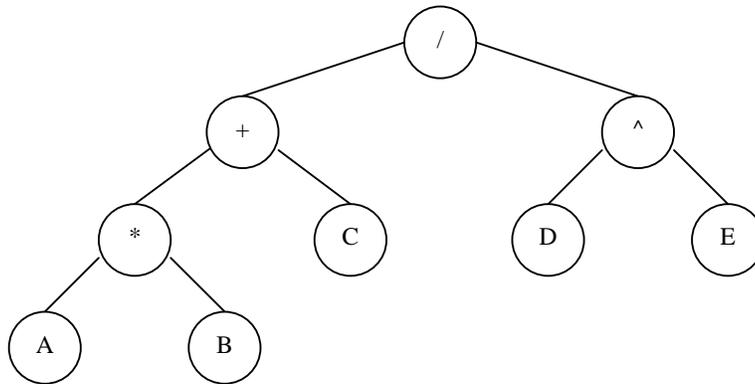
Expression Tree

- Simpul daun : operand.
- Simpul cabang : operator.

Pandang ekspresi matematik berikut :

$$(A * B + C) / (D ^ E)$$

Pohon biner untuk ungkapan aritmetik :



Preorder atau notasi prefix : / + * A B C ** D E
Postorder atau notasi postfix : A B * C + D E ** /
Inorder atau notasi infix : A * B + C / D ** E
Level order : / + ** * C D E A B