PERTEMUAN 13
Tree

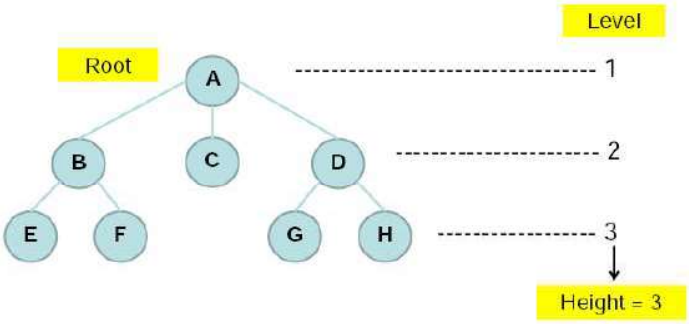
8.1 Pengertian Tree (Pohon)

Pohon (Tree) termasuk struktur non linear yang didefinisikan sebagai data yang terorganisir dari suatu item informasi cabang yang saling terkait

8.2 Isilah Dasar dalam Tree

Pohon atau Tree adalah salah satu bentuk Graph terhubung yang tidak mengandung sirkuit. Karena merupakan Graph terhubung, maka pada Pohon (Tree) selalu terdapat Path atau Jalur yang menghubungkan setiap simpul dalam dua pohon. Pohon (Tree) dapat juga didefinisikan sebagai kumpulan elemen yang salah satu elemennya disebut dengan Akar (Root) dan sisa elemen lain (Simpul) yang terpecah menjadi sejumlah himpunan yang saling tidak berhubungan yang disebut dengan Subpohon (Subtree) atau cabang

8.3 Isilah-Istilah dalam Tree



Gambar 8.1 Contoh Tree

1. Predesesor

   Node yang berada diatas node tertentu. (contoh : B predesesor dari E dan F)

2. Succesor

   Node yang berada dibawah node tertentu. (contoh : E dan F merupakan succesor dari B)

3. Ancestor

   Seluruh node yang terletak sebelum node tertentu dan terletak pada jalur yang sama. (contoh : A dan B merupakan ancestor dari F)

4. Descendant

   Seluruh node yang terletak sesudah node tertentu dan terletak pada jalur yang sama. (contoh : F dan B merupakan ancestor dari A)

5. Parent

   Predesesor satu level diatas satu node (contoh : B merupakan parent dari F)

6. Child

   Succesor satu level dibawah satu node (contoh : F merupakan child dari B)

7. Sibling

   Node yang memiliki parent yang sama dengan satu node (contoh : E dan F adalah sibling)

8. Subtree

   Bagian dari tree yang berupa suatu node beserta descendant-nya (contoh : Subtree B, E, F dan Subtree D, G, H)

9. Size

   Banyaknya node dalam suatu tree (contoh : gambar tree diatas memiliki size = 8)

10. Height

    Banyaknya tingkat/level dalam suatu tree (contoh : gambar tree diatas memiliki height = 3)

11. Root (Akar)

    Node khusus dalam tree yang tidak memiliki predesesor (Contoh : A)

12. Leaf (Daun)

    Node-node dalam tree yang tidak memiliki daun (contoh : Node E,F,C,G,H)

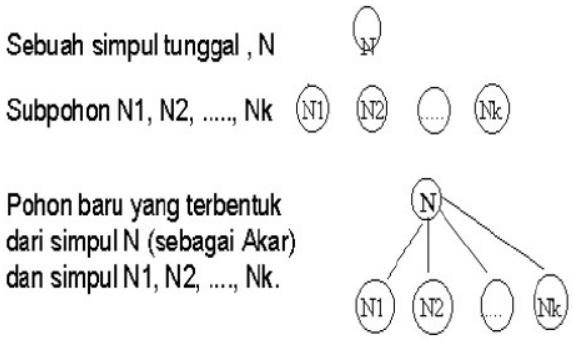
13. Degree (Derajat)

    Banyaknya child yang dimiliki oleh suatu node (contoh : Node A memiliki derajat 3, node B memiliki derajat 2)

8.4 Sifat Utama Tree Berakar

Pembentukan Tree

Gambar berikut ini menjelaskan tentang pembentukan awal dari Pohon (Tree).



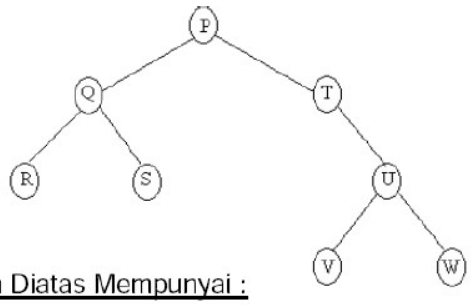
1. Jika Pohon mempunyai Simpul sebanyak n, maka banyaknya ruas atau edge adalah (n-1).
2. Mempunyai Simpul Khusus yang disebut Root, jika Simpul tersebut memiliki derajat keluar >= 0, dan derajat masuk = 0.
3. Mempunyai Simpul yang disebut sebagai Daun / Leaf, jika Simpul tersebut berderajat keluar = 0, dan berderajat masuk = 1.
4. Setiap Simpul mempunyai Tingkatan / Level yang dimulai dari Root yang Levelnya = 1 sampai dengan Level ke – n pada daun paling bawah. Simpul yang mempunyai Level sama disebut Bersaudara atau Brother atau Stribling.
5. Pohon mempunyai Ketinggian atau Kedalaman atau Height, yang merupakan Level tertinggi
6. Pohon mempunyai Weight atau Berat atau Bobot, yang banyaknya daun (leaf) pada Pohon.
7. Banyaknya Simpul Maksimum sampai Level N adalah :

$$2^{(N)} - 1$$

8. Banyaknya Simpul untuk setiap Level I adalah :

$$\sum_{I=1}^{N} 2^{(I-1)}$$

Hutan (Forest) adalah kumpulan Pohon yang tidak saling berhubungan. Diketahui suatu bentuk Pohon Berakar T sebagai berikut :



Pohon Diatas Mempunyai :

a. Simpul sebanyak = 8 dan edge = n - 1 = 8 – 1 = 7
b. Root pada Pohon T diatas adalah Simpul P
c. Mempunyai daun (Leaf) = 4, yaitu = R, S, V dan W

d. Level (tingkatan) Pohon = 4 yaitu :

   Level 1 = Simpul P

   Level 2 = Simpul Q dan T

   Level 3 = Simpul R, S dan U

   Level 4 = Simpul V dan W

e. Ketinggian atau kedalaman = jumlah level = 4

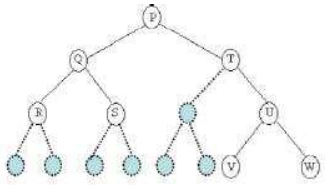
f. Weight atau berat atau bobot = jumlah daun = 4

   Dalam gambar Pohon T diatas dapat dibentuk 2 buah hutan (forest), bila simpul P dihilangkan, yaitu : Hutan 1 : Q,R,S Hutan 2 : T,U,V,W

g. Banyaknya Simpul Maksimum yang dapat terbentuk sampai Level 4 (bila simpul pada pohon dianggap penuh) adalah :

   $2^{(N)} - 1$

   $2^{(4)} - 1 = 16 - 1 = 15$



Gambar 8.2 Banyak Simpul

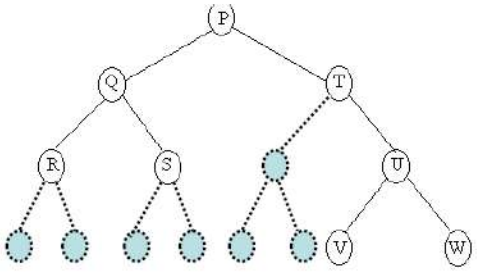
h. Banyaknya Simpul maksimum untuk setiap Level I (bila simpul pada pohon dianggap penuh) adalah :

Maksimum Simpul pada level 2 = $\mathbf{2^{(I-1)}}$

$= 2^{(2-1)} = 2$

Maksimum Simpul pada level 3 = $2^{(3-1)} = 4$

Maksimum Simpul pada level 4 = $2^{(4-1)} = 2$



Gambar 8.3 Banyak Simpul setiap Level

8.5 Pohon Biner (Binary Tree)

Struktur ini biasanya digunakan untuk menyajikan data yang mengandung hubungan hirarkial antara elemenelemennya. Bentuk Pohon Berakar yang lebih mudah dikelola dalam komputer adalah Pohon Biner (Binary Tree) yang lebih dikenal sebagai Pohon Umum (General Tree) yang dapat didefinisikan sebagai kumpulan simpul yang mungkin kosong atau mempunyai akar dan dua Subpohon yang saling terpisah yang disebut dengan Subpohon Kiri / cabang kiri (Left Subtree) dan Subpohon Kanan / cabang kanan (Right Subtree).

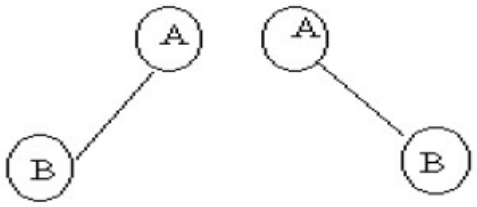
Karakteristik Pohon Binar (Binary Tree) :

1. Setiap Simpul paling banyak hanya memiliki dua buah anak
2. Derajat Tertinggi dari setiap Simpul adalah dua.

3. Dibedakan antara Cabang Kiri dan Cabang Kanan.

4. Dimungkinkan tidak mempunyai Simpul

Berikut ini diberikan contoh gambar Pohon Binar (Binary Tree) dengan *Cabang Kiri dan Cabang Kanan*.

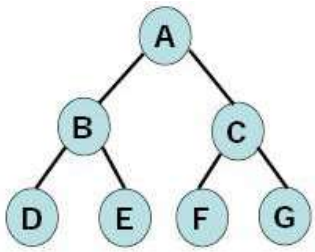


Gambar 8.4 Cabang Kiri dan Cabang Kanan

8.6 Istilah-istilah Pada Pohon Biner

1. **Pohon Biner Penuh (Full Binary Tree)**

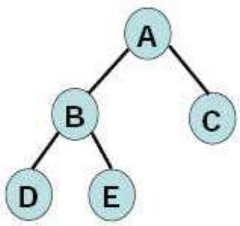
Semua simpul (kecuali daun) memiliki 2 anak dan tiap cabang memiliki panjang ruas yang sama



Gambar 8.5 Pohon Biner Penuh

2. **Pohon Biner Lengkap (Complete Binary Tree)**

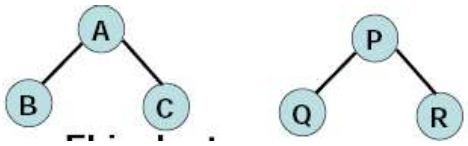
Hampir sama dengan Pohon Biner Penuh, semua simpul (kecuali daun) memiliki 2 anak tetapi tiap cabang memiliki panjang ruas berbeda.



Gambar 8.6 Pohon Biner Lengkap

3. **Pohon Biner Similer**

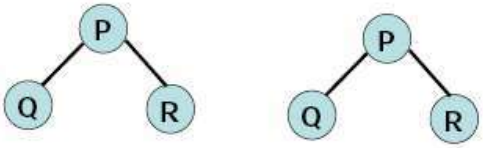
Dua pohon yang memiliki struktur yang sama tetapi informasinya berbeda.



Gambar 8.7 Pohon Biner Similer

4. **Pohon Biner Ekivalent**

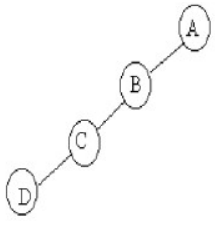
Dua pohon yang memiliki struktur dan informasi yang sama.



Gambar 8.8 Pohon Biner Ekivalent

5. **Pohon Biner Miring (Skewed Tree)**

Dua pohon yang semua simpulnya mempunyai satu anak / turunan kecuali daun



Gambar 8.9 Pohon Biner Miring

8.7 Deklarasi Pohon Biner

Dalam setiap simpul selalu berisi dua buah Pointer untuk menunjuk ke cabang Kiri dan cabang Kanan dan informasi yang akan disimpan dalam simpul tersebut.

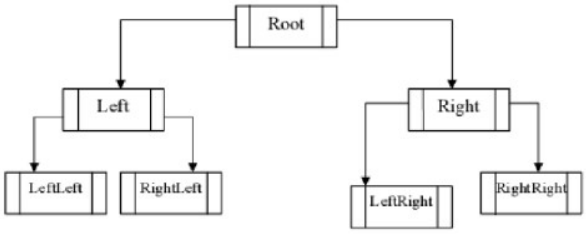
Deklarasi struct

```
typedef struct Tree{
                    int data;
                    Tree *left;
                    Tree *right;
                }
```

Deklarasi variabel:

```
Tree *pohon;
```

Ilustrasi Pohon Biner



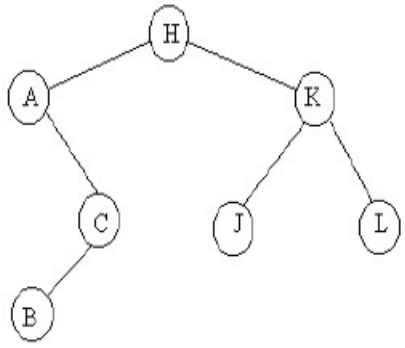
Gambar 8.10 Ilustrasi Pohon Biner

8.8 Penyajian Pohon Biner

Tree dapat dibuat dengan menggunakan linked list secara rekursif. Linked list yang digunakan adalah double linked list non circular. Data yang pertama kali masuk akan menjadi node root. Data yang lebih kecil dari data node root akan masuk dan menempati node kiri dari node root, sedangkan jika lebih besar dari data node root, akan masuk dan menempati node di sebelah kanan node root. Bila diberikan untai HAKJCBL, maka proses untuk dapat membentuk pohon biner dari untai diatas adalah :

1. Karakter pertama ‘H’ ditempatkan sebagai akar (root)
2. Karakter ‘A’,karena lebih kecil dari ‘H’, maka akan menempati cabang kiri.
3. Karakter ‘K’, karena lebih besar dari ‘H’, maka akan menempati cabang kanan.
4. Karakter ‘J’, lebih besar dari ‘H’ dan kecil dari ‘K’, maka menempati cabang kiri ‘K’.
5. Karakter ‘C’,karena lebih besar dari ‘A’, maka akan menempati cabang kanan.
6. Karakter ‘B’, karena lebih kecil dari ‘C’, maka akan menempati cabang kiri.
7. Karakter ‘L’, lebih besar dari ‘K’, maka menempati cabang kiri kanan.

Sehingga terbentuk pohon biner seperti berikut :



Gambar 8.11 Contoh Penyajian Pohon

C. SOAL LATIHAN/TUGAS PENDAHULUAN

Latihan 13

1. Apa yang dimaksud dengan Algoritma Tree ?
2. Sebutkan istilah-istilah pada pohon biner ?

LAPORAN AKHIR PERTEMUAN 13 (Dikumpulkan pada Pertemuan 14)

Tugas Akhir

1. Buatlah program dengan menggunakan Algoritma Tree

Laporan Akhir

1. Jelaskan program yang telah anda buat pada Tugas Akhir dengan detail dan jelas