Pertemuan 13

Waktu : 135 menit

Tujuan Pembelajaran : Mahasiswa mampu menjelaskan teknik pemrograman menggunakan Tree.

Substansi Materi : Tree

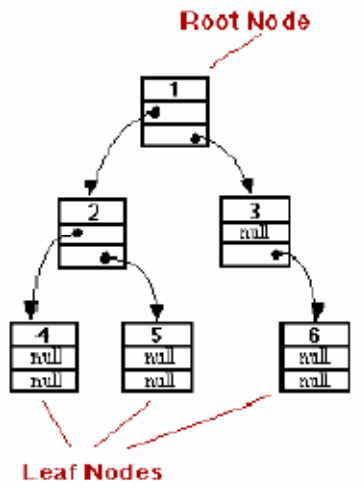
Tabulasi Kegiatan Perkuliahan

No	Tahap Kegiatan	Kegiatan Pengajar	Kegiatan Mahasiswa	Media & Alat	Waktu
1	Pendahuluan	1. Membuka pertemuan 2. Mengulang materi pertemuan sebelumnya	Menyimak Bertanya	Papan Tulis	20 Menit
2	Penyajian Materi	1. Pengertian Tree 2. Jenis Tree 3. Binary Tree 4. Contoh program 5. Contoh soal Tree	Menyimak Bertanya Menjawab Pertanyaan	Papan Tulis	80 Menit
3	Penutup	1. Menyimpulkan materi pertemuan 2. Memberikan tugas kecil 3. Menutup pertemuan	Menyimak	Papan tulis	35 Menit

MATERI KULIAH

TREE

Sebelumnya kita sudah mengenal struktur data list, yang berupa obyek-obyek yang saling terkait. Dalam list, satu obyek hanya terkait dengan satu obyek berikutnya melalui sebuah pointer. List dapat dikembangkan menjadi struktur data yang lebih kompleks, misalnya dengan menambah jumlah pointer dalam obyek. Misal dengan penambahan satu pointer lagi. Artinya bahwa jika masing-masing obyek memiliki dua pointer, ada dua obyek lain yang ditunjuknya. Struktur yang demikian dikenal sebagai binary tree atau dikenal juga sebagai Tree Node.

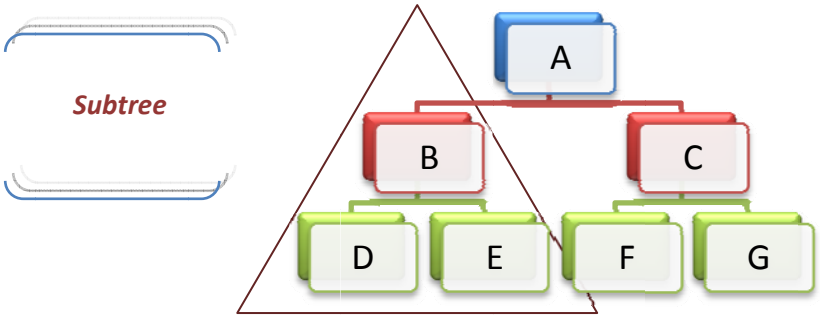


Gambar 1. Ilustrasi Binary Tree

Istilah-istilah umum dalam Binary Tree :

- Predecessor : node yang berada di atas 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 sesudah node tertentu dan terletak pada jalur yang sama
- Parent : predecessor satu level diatas suatu node
- Child : successor satu level diatas suatu node
- Subtree : bagian dari tree yang berupa suatu node beserta descendantnya dan memiliki semua karakteristik dari tree tersebut
- Size : Banyaknya node dalam suatu tree
- Height : Banyaknya tingkatan / level dalam suatu tree
- Root : Satu-satunya node khusus dalam tree yang tak punya predecessor
- Leaf : Node-node dalam tree yang tak memiliki successor
- Degree : Banyaknya child yang dimiliki suatu node

Contoh :



Jenis-jenis Tree

1. Binary Tree

   Binary Tree adalah tree dengan syarat bahwa tiap node hanya boleh memiliki maksimal dua subtree dan kedua subtree tersebut harus terpisah. Sesuai dengan definisi tersebut, maka tiap node dalam binary tree hanya boleh memiliki paling banyak dua child.

   Jenis-jenis Binary Tree :

   - Full Binary Tree

     Binary Tree yang tiap nodenya (kecuali leaf) memiliki dua child dan tiap subtree harus mempunyai panjang path yang sama.

   - Complete Binary Tree

     Mirip dengan Full Binary Tree, namun tiap subtree boleh memiliki panjang path yang berbeda. Node kecuali leaf memiliki 0 atau 2 child.

   - Skewed Binary Tree

     Yakni Binary Tree yang semua nodenya (kecuali leaf) hanya memiliki satu child.

Deklarasi Binary Tree

```
Type   Tree = ^node;
       Node = record
              Isi : TipeData;
              Left, Right : Tree;
       End;
```

Operasi-operasi pada Binary Tree

- Create : Membuat binary tree baru yang masih kosong
- Clear : Mengosongkan binary tree yang sudah ada
- Empty : Function untuk memeriksa apakah binary tree masih kosong.
- Insert : Memasukan sebuah node ke dalam tree. Ada tiga pilihan insert, yaitu ROOT, LEFT CHILD, atau RIGHT CHILD. Khusus insert sebagai ROOT, TREE harus dalam keadaan kosong.
- Find : Mencari root, parent, left child, atau right child dari suatu node. Tree tidak boleh dalam kedaan kosong.
- Update : Mengubah isi dari node yang ditunjuk oleh pointer current. Tree tidak boleh dalam keadaan kosong.
- Retrieve : Mengetahui isi dari node yang ditunjuk oleh pointer kosong. Tree tidak boleh dalam kedaan kosong.
- DeleteSub : Menghapus sebuah subtree (node beserta seluruh descendant-nya) yang ditunjuk oleh current. Tree tidak boleh kosong. Setelah itu pointer current akan berpindah ke parent dari node yang telah di hapus.
- Characteristic: Mengetahui karakteristik dari suatu tree, yakni : size, height, serta average dari length-nya. Tree tidak boleh kosong.

- Traverse : Mengunjungi seluruh node-node pada tree, masing-masing sekali. Hasilnya adalah urutan informasi secara linier yang tersimpan dalam tree. Ada tiga cara traverse : Pre Order, In Order, dan Post Order.

  Langkah melakukan traverse :

  - PreOrder : cetak isi node yang dikunjungi, kunjungi Left Child, kunjungi Right Child.
  - InOrder : Kunjungi Left Child, cetak isi node yang dikunjungi, kunjungi Right Child.
  - PostOrder : Kunjungi Left Child, kunjungi Right Child, cetak isi node yang dikunjungi.