

Praktikum Struktur Data

Pertemuan Ke-10

TREE

- Struktur pada tree (pohon) tidak linear seperti pada struktur linked list, stack, dan queue.
- Setiap node pada tree mempunyai tingkatan, yaitu orang tua (parent) dan anak (child).
- Struktur tree merupakan bentuk khusus dari struktur tree yang lebih umum, setiap orang tua hanya memiliki dua anak sehingga disebut pohon biner (binary tree), yaitu anak kiri dan anak kanan.

Operasi Pada Tree

```
Function BARU(Hrf : Char) : Tree;  
Var Temp : Tree;  
Begin  
    New(Temp);  
    Temp^.Info := Hrf;  
    Temp^.Kiri := NIL; Temp^.Kanan := NIL;  
    BARU := Temp;  
End;
```

Operasi Pada Tree

```
Procedure MASUK(Var Pohon : Tree; Hrf : Char);  
Begin  
If Pohon = NIL Then  
Pohon := BARU(Hrf)  
Else  
Begin  
If Pohon^.Info > Hrf then  
MASUK(Pohon^.Kiri,Hrf)  
Else If Pohon^.Info < Hrf then  
MASUK(Pohon^.Kanan,Hrf)  
Else  
Writeln('Karakter', Hrf, 'Sudah ada di Tree');  
End;  
End;
```

Binary Tree

- Binary tree adalah 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.

Implementasi Binary Tree

Binary Tree dapat diimplementasikan dalam Pascal dengan menggunakan double Linked List. Untuk nodenya, dapat dideklarasikan sbb :

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

Contoh : Program Tree

```
1 uses wincrt;
2 Type
3     Tree = ^Simpul;
4     Simpul = Record
5         Info : char;
6         Kiri : Tree;
7         Kanan : Tree;
8     End;
9
10 Function BARU(Hrf : Char) : Tree;
11 Var Temp : Tree;
12 Begin
13     New(Temp);
14     Temp^.Info := Hrf;
15     Temp^.Kiri := NIL; Temp^.Kanan := NIL;
16     BARU := Temp;
17 End;
```

Contoh : Program Tree

```
19 Procedure MASUK(Var Pohon : Tree; Hrf : Char);  
20 Begin  
21     If Pohon = NIL Then  
22         Pohon := BARU(Hrf)  
23     Else  
24         Begin  
25             If Pohon^.Info > Hrf then  
26                 MASUK(Pohon^.Kiri,Hrf)  
27             Else If Pohon^.Info < Hrf then  
28                 MASUK(Pohon^.Kanan,Hrf)  
29             Else  
30                 Writeln('Karakter', Hrf, 'Sudah ada di Tree');  
31         End;  
32     End;
```

Contoh : Program Tree

```
33
34 Procedure PREORDER(Temp : Tree);
35 Begin
36     If Temp <> NIL Then
37         Begin
38             Write(Temp^.Info, ' ');
39             PREORDER(Temp^.Kiri);
40             PREORDER(Temp^.Kanan);
41         End;
42     End;
43
44 Procedure INORDER(Temp : Tree);
45 Begin
46     If Temp <> NIL Then
47         Begin
48             INORDER(Temp^.Kiri);
49             Write(Temp^.Info, ' ');
50             INORDER(Temp^.Kanan);
51         End;
52     End;
```

Contoh : Program Tree

```
53
54 Procedure POSTORDER(Temp : Tree);
55 Begin
56     If Temp <> NIL Then
57         Begin
58             POSTORDER(Temp^.Kiri); {Kunjungi cabang kiri}
59             POSTORDER(Temp^.Kanan); {Kunjungi cabang kanan}
60             Write(Temp^.Info, ' '); {Cetak isi simpul}
61         End;
62     End;
```

Contoh : Program Tree

```
63
64 var
65 poon:tree;
66 Begin
67 clrscr;
68     MASUK(poon, 'b');
69     MASUK(poon, 'c');
70     MASUK(poon, 'u');
71     MASUK(poon, 'e');
72     MASUK(poon, 'a');
73     writeln('PERORDER:');
74     PREORDER(poon);
75     writeln;
76     writeln('INORDER:');
77     INORDER(poon);
78     writeln;
79     writeln('POSTORDER:');
80     POSTORDER(poon);
81     readln;
82 end.
```

Thank you!

