

# Praktikum Struktur Data

Pertemuan Ke-11

# BINARY SEARCH TREE (BST)

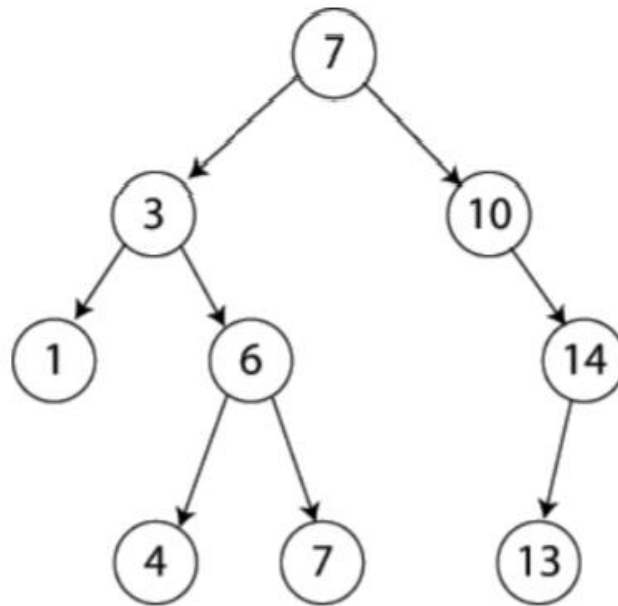
- ❑ Binary Search Tree (BST) merupakan tree yang terurut (ordered Binary Tree) yang memiliki kelebihan bila dibanding dengan struktur data lain.
- ❑ Diantaranya adalah proses pengurutan (sorting) dan pencarian (searching) dapat dilakukan bila data sudah tersusun dalam struktur data BST.
- ❑ Pengurutan dapat dilakukan bila BST ditelusuri (traversed) menggunakan metode in-order

# Aturan dalam membentuk BST

- ❑ Semua data dibagian kiri sub-tree dari node  $t$  selalu lebih kecil dari data dalam node  $t$  itu sendiri. 2.
- ❑ Semua data dibagian kanan sub-tree dari node  $t$  selalu lebih besar atau sama dengan data dalam node  $t$ .

# Contoh BST

BST berikut adalah sebuah BST berukuran 9 dengan kedalaman 3 dengan node daun (leaf) adalah 1, 4, 7 dan 13.



# Aplikasi BST

- ❑ Membangun daftar vocabulary yang merupakan bagian dari inverted index (sebuah struktur data yang digunakan oleh banyak mesin pencari seperti Google.com, Yahoo.com dan Ask.com)
- ❑ Banyak digunakan dalam bahasa pemrograman untuk mengelola dan membangun dynamic sets.

# Contoh: Program BST

```
1 program AplikasiBST;
2 uses crt;
3
4 type
5 tipeinfo=char;
6 tree=^simpul;
7 simpul = record
8     info : tipeinfo;
9     kiri,kanan : tree;
10 end;
11
12 var kata:string;
13     pohon : tree;
14     i: byte;
15
16
```

# Contoh : Program BST

```
17 procedure inisialisasi(var pohon: tree);
18 begin
19   new(pohon);
20   pohon:=nil;
21 end;
22
23 procedure insertdata(var pohon:tree;data:tipeinfo);
24 var baru :tree;
25 begin
26   new(baru);
27   baru^.info:=data;
28   baru^.kiri:=nil;
29   baru^.kanan:=nil;
30   if(pohon=nil) then pohon:= baru
31   else if (data <>
32     else insertdata(pohon^.kanan,data);
33 end;
34
```

# Contoh : Program BST

```
35 procedure prefix(pohon :tree);
36 begin
37   if pohon<>nil then begin
38     write(pohon^.info);
39     prefix(pohon^.kiri);
40     prefix(pohon^.kanan); end;
41   end;
42 procedure infix(pohon :tree);
43 begin
44   if pohon<>nil then begin
45     infix(pohon^.kiri);
46     write(pohon^.info);
47     infix(pohon^.kanan); end;
48   end;
49
50 procedure posfix(pohon :tree);
51 begin
52   if pohon<> nil then begin
53     posfix(pohon^.kiri);
54     posfix(pohon^.kanan);
55     write(pohon^.info); end;
56   end;
```



# Contoh: Program BST

```
58 begin
59 write('masukkan kata');
60 readln(kata); inisialisasi(pohon);
61 for i:=1 to length(kata) do
62   insertdata(pohon, kata[i]);
63 writeln('prefix');
64 prefix(pohon);
65 writeln;
66 writeln('infix');
67 infix(pohon);
68 writeln;
69 writeln('posfix');
70 posfix(pohon);
71 writeln;
72 readln;
73 end.
```

Thank you!

