Nama : Yudhi Dwi Tananjaya

NIM : 18.N1.0011

Tugas Struktur Data

1. Public BST

Untuk menambahkan nama yang akan dibuat kolom.

Contohnya:

Class BST {

Private Node root;

Public booleanisempty()

{

return root==null;

}

public void insert(int i) { }

}

1. Boolean Isempty

Fungsi ini untuk menampilkan nilai kosong atau tidak.

1. Void insert

Fungsi ini untuk menambahkan data yang akan ditambahkan.

Contohnya

BST a = new BST();

a.insert(3);

jadi fungsi ini dapat memastikan akan masuk ke sebelah kiri atau sebelah kanan, supaya fungsi ini bekerja maka kita menggunakan fungsi ini:

} else {

while(x!=null) {

y=x;

if(i<x.data)

x=x.left;

else

x=x.right;

}

if(i<y.data) {

y.left=temp;

System.out.println("nilai"+i+" masuksebelahkiri"+y.data);

} else {

y.right=temp;

System.out.println("nilai"+i+" masuksebelahkanan"+y.data);

}

Dengan menggunakan fungsi ini a.insert(3) dapat ditentukan akan masuk di sebelah kiri atau kanan.



1. Void findmin

Fungsi ini untuk memunculkan nilai minimal dari angka tree tersebut.

Contohnya:

public void findmin(){

node temp=root;

while(temp.left!=null){

temp=temp.left;

}

System.out.println("nilaiminimum :"+temp.data);

}

Maka setelah di run akan muncul “nilai minimum : 1“.



1. Void findmax

Fungsi ini untuk memunculkan nilai maximal dari angka tree tersebut

Contohnya:

public void findmax(){

node temp=root;

while(temp.right!=null){

temp=temp.right;

}

System.out.println("nilaimaximum :"+temp.data);

}

Maka setelah di run akan muncul “nilai maximal : 16”.



1. Void Cekroot

Fungsi ini untuk menghitung jumlah data.

Contohnya:

public void cekroot(){

System.out.println("nilairoot :"+root.data);

}

Maka setelah di run akan muncul “nilai root : 2” maksud mempunyai 2 data.



1. Void urutroot

Fungsi ini untuk mengurutkan Tree dari nilai terendah sampai nilai tertinggi.

Contohnya:

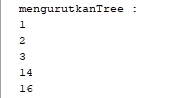
public void urutroot(){

System.out.println("mengurutkanTree :");

uruttree(root);

}

Maka setelah di run akan muncul ”mengurutkan Tree : 1 2 3 14 16”.



1. Void delete

Fungsi ini untuk menghapus nilai tree. Hasilnya setelah di jalankan/run akan muncul “nilai 2 telah dihapus ”



1. Void searching

Fungsi ini untuk mencari nilai tree yang masih ada di data tersebut. Hasilnya setelah di run akan muncul “data ditemukan : 16”



1. Void transplanted

Fungsi ini untuk membantu fungsi delete untuk memisahkan angka yang akan dihapus dengan angka yang tidak ingin dihapus.