

JOBSHEET 14
BINARY TREE

NAMA : DWI NUR OKTAVIANI
NIM (ABSEN) : 1941720239 (09)
KELAS : TI – 1F
TANGGAL PRAKTIKUM : 14 MEI 2020

PERTANYAAN :

1. Mengapa dalam Binary search tree proses pencarian data bisa lebih efektif dilakukan dibanding binary tree biasa?

Jawab :

Karena harus membedakan child kanan sesuai besaran nilainya, atau membagi dengan mencari 2 arah jika struktur data tree sudah tersusun rapi sesuai aturan mainnya, proses search akan lebih cepat.

2. Untuk apakah di class Node, kegunaan dari atribut left dan right?

Jawab :

Di Class Node atribut left digunakan untuk menyimpan data disebelah kiri dan atribut rightt menyimpan data sebelah kanan

3. a. Untuk apakah kegunaan dari atribut root di dalam class BinaryTree?

Jawab :

Atribut pada class BinaryTree sebagai data awal atau sebagai akar

- b. Ketika objek tree pertama kali dibuat, apakah nilai dari root?

Jawab :

Nilai dari root = Null

4. Ketika tree masih kosong, dan akan ditambahkan sebuah node baru, proses apa yang akan terjadi?

Jawab :

-Node baru yang ditambahkan menempati posisi root

-Root = new node(data)



JOBSHEET 14
BINARY TREE

NAMA : DWI NUR OKTAVIANI
NIM (ABSEN) : 1941720239 (09)
KELAS : TI – 1F
TANGGAL PRAKTIKUM : 14 MEI 2020

5. Perhatikan method `add()`, di dalamnya terdapat baris program seperti di bawah ini. Jelaskan secara detil untuk apa baris program tersebut?

```
if(data<current.data){  
    if(current.left!=null){  
        current = current.left;  
    }else{  
        current.left = new Node(data);  
        break;  
    }  
}
```

Jawab :

Memasukkan suatu nilai ke dalam tree, jika nilai tersebut tidak kosong maka bergerak terus ke kiri hingga menjadi child, begitu juga sebaliknya

6. Apa perbedaan antara traverse mode pre-order, in-order dan post-order?

Jawab :

Pre-order : cetak isi node yang dikunjungi, kunjungi left son, kunjungi right son

In-order : kunjungi left son, cetak isi node yang dikunjungi, kunjungi right son

Post-Order : kunjungi left son, kunjungi right son, cetak isi node yang dikunjungi.

7. Perhatikan method **`delete()`**. Sebelum proses penghapusan node, didahului dengan proses pencarian node yang akan dihapus. Selain ditujukan untuk mencari node yang akan dihapus (**current**), proses pencarian tersebut juga akan mencari parent dari node yang akan dihapus (**parent**). Menurut Anda, mengapa diperlukan juga untuk mengetahui parent dari node yang akan dihapus?

Jawab :

Diperlukan karena selama sebelum menemukan nilai yang sama, jika nilai yang dicari lebih kecil dari nilai yang disimpan dalam Node maka bergerak ke left child begitu juga sebaliknya



JOBSHEET 14
BINARY TREE

NAMA : DWI NUR OKTAVIANI
NIM (ABSEN) : 1941720239 (09)
KELAS : TI – 1F
TANGGAL PRAKTIKUM : 14 MEI 2020

8. Untuk apakah dibuat variabel dengan nama **isLeftChild** di dalam method **delete()**?

Jawab :

Untuk menandai apakah Node yang dimaksud isLeftChild

9. Untuk apakah method **getSuccessor()**?

Jawab :

Method ini akan digunakan ketika proses penghapusan node yang memiliki 2 child.

10. Di ulasan teori, disebutkan bahwa ketika suatu node yang memiliki 2 child dihapus, node tersebut digantikan oleh node successor, dimana node successor bisa didapat dengan 2 cara, yaitu:

1. mencari nilai terbesar dari subtree di sebelah kirinya, atau
- mencari nilai terkecil dari subtree di sebelah kanannya.

Manakah 1 dari 2 cara tersebut yang diimplementasikan dalam method **getSuccessor()** di program di atas?

Jawab :

Mencari nilai terkecil dari subtree di sebelah kanannya

11. Apakah kegunaan dari atribut data dan idxLast yang ada di class **BinaryTreeArray**?

Jawab :

Atribut idxLast digunakan untuk menyimpan index terakhir dari array

12. Apakah kegunaan dari method **populateData()** dan **traverseInOrder()**?

Jawab :

-populateData() : untuk mendeklarasikan data yang diambil dari class main dan dimasukkan ke class binaryTreeArray

-traverseInOrder : digunakan untuk mengunjungi left son, mencetak isi node yang dikunjungi, mengunjungi right son.



JOBSHEET 14
BINARY TREE

NAMA : DWI NUR OKTAVIANI
NIM (ABSEN) : 1941720239 (09)
KELAS : TI – 1F
TANGGAL PRAKTIKUM : 14 MEI 2020

13. Jika suatu node binary tree disimpan dalam array indeks 2, maka di indeks berapakah posisi left-child dan right child masing-masing?

Jawab :

-leftChild : $2*2+1 = 5$

-rightChild : $2*2+2 = 62$

