**ALGORITMA STRUKTUR DATA**

**Praktikum - Graph**

**Lavina 2341760062**

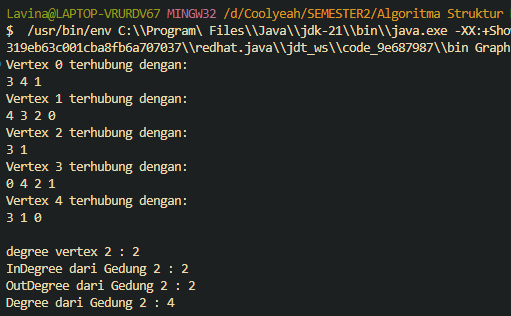
**Praktikum 1**

Graph.java



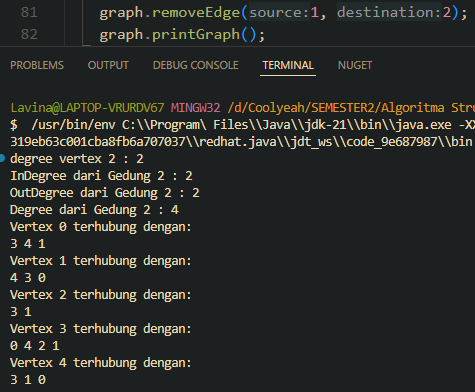


Output :



Tambahkan pemanggilan method **removeEdge()** sesuai potongan code di bawah ini pada method main(). Kemudian tampilkan graph tersebut.





1. Sebutkan beberapa jenis (minimal 3) algoritma yang menggunakan dasar Graph, dan apakah kegunaan algoritma-algoritma tersebut?

**Jawab :**

* 1. Algoritma Dijkstra

Kegunaan: Algoritma Dijkstra digunakan untuk menemukan jalur terpendek dari satu simpul ke semua simpul lain dalam graf berbobot tanpa bobot negatif. Ini sering digunakan dalam aplikasi seperti perencanaan rute, navigasi GPS, dan analisis jaringan komputer.

* 1. Algoritma Kruskal

Kegunaan: Algoritma Kruskal digunakan untuk menemukan Minimum Spanning Tree (MST) dalam graf berbobot. MST adalah pohon yang menghubungkan semua simpul dalam graf dengan total bobot tepi minimum. Ini berguna dalam masalah seperti jaringan kabel telekomunikasi, pengurangan biaya konstruksi jaringan, dan optimisasi distribusi.

* 1. Breadth-First Search (BFS)

Kegunaan: BFS digunakan untuk menjelajahi graf atau pohon secara luas, dimulai dari akar atau simpul yang diberikan, dan mengunjungi semua simpul di tingkat saat ini sebelum beralih ke simpul di tingkat berikutnya. Ini berguna dalam pencarian rute terpendek dalam graf tak berbobot, penyelesaian teka-teki, dan pemodelan penyebaran penyakit dalam jaringan sosial.

1. Pada class Graph terdapat array bertipe LinkedList, yaitu LinkedList list[]. Apakah tujuan pembuatan variabel tersebut ?

**Jawab :** Fungsinya untuk menyimpan data vertex dan adjency list dalam graph.

1. Apakah alasan pemanggilan method **addFirst()** untuk menambahkan data, bukan method add jenis lain pada linked list ketika digunakan pada method addEdge pada **class Graph?**

**Jawab :** Karena method itu adalah cara paling cepat untuk menambahkan data pada graph tanpa melakukan traverse.

1. Bagaimana cara mendeteksi prev pointer pada saat akan melakukan penghapusan suatu edge pada graph ?

**Jawab :**

1. Kenapa pada praktikum 2.1.1 langkah ke-12 untuk menghapus path yang bukan merupakan lintasan pertama kali menghasilkan output yang salah ? Bagaimana solusinya ?

**Pertanyaan**

* 1. Mengapa dalam binary search tree proses pencarian data bisa lebih efektif dilakukan dibanding binary tree biasa?
  2. Untuk apakah di class **Node**, kegunaan dari atribut **left** dan **right**?

**Jawab :** Gunanya sama dengan kegunaan pointer dalam double linked list, left menunjuk pada child kiri dan right menunjuk pada child kanan.

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

**Jawab :** Sebagai node yang berada paling atas dan tidak memiliki parent.

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

**Jawab :** Kondisi awal **root** berisi nilai **null**

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

**Jawab :** Ketika masih kosong maka node root akan diisi dengan data baru yang ingin ditambahkan.

* 1. 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;**

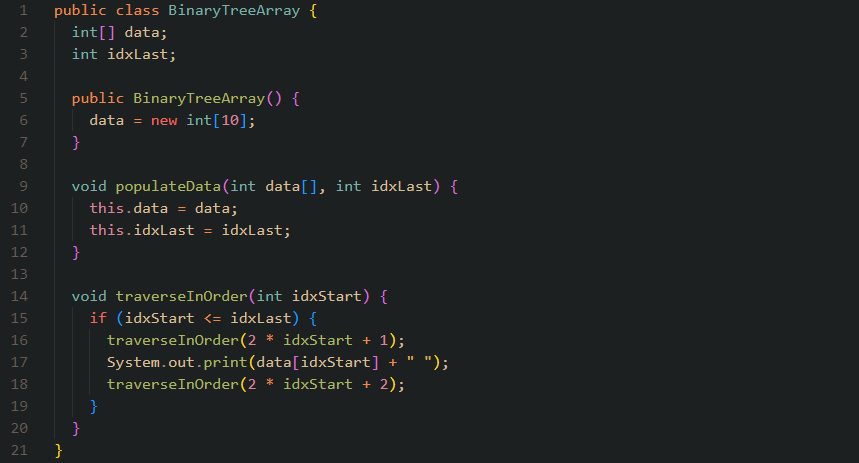
**}**

**}**

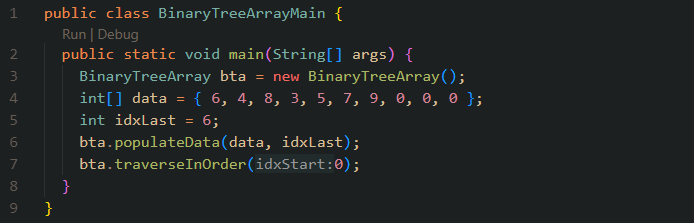
**Jawab :** Pertama mengecek apakah data yang ingin ditambah lebih kecil dengan data saat ini, jika true maka dilakukan pengecekan apakah child kiri dari current kosong atau tidak, jika kosong maka current saat ini dipindah ke sebelah kiri tersebut sampai current left nya bernilai null, ketika sudah null diisi dengan data yang baru ingin ditambahkan.

**Percobaan 2**

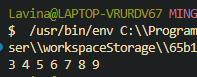
BinaryTreeArray.java

****

BinaryTreeArrayMain.java



Output :



**Pertanyaan**

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

**Jawab :** Atribut data berfungsi untuk menyimpan array dari node-node pada tree, sedangkan idxLast adalah variabel untuk menyimpal value index terakhir dalam tree.

* 1. Apakah kegunaan dari method **populateData()**?

**Jawab :** Method tersebut berguna sebagai konstruktor yang mengatur kondisi awal dari array data dan isi dari variable idxLast yang didapat melalui parameter.

* 1. Apakah kegunaan dari method **traverseInOrder()**?

**Jawab :** Berguna untuk menelusuri tree secara dengan metode in order.

* 1. Jika suatu node binary tree disimpan dalam array indeks 2, maka di indeks berapakah posisi left child dan rigth child masin-masing?

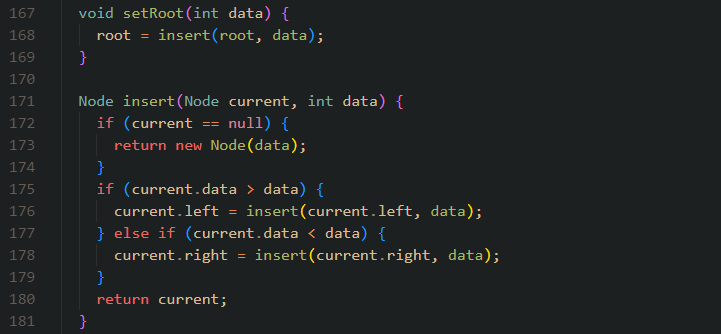
**Jawab :** Posisi left child = 2 \* (2) + 1 = 5 dan posisi right child = 2 \* (2) + 2 = 6

* 1. Apa kegunaan statement int idxLast = 6 pada praktikum 2 percobaan nomor 4?

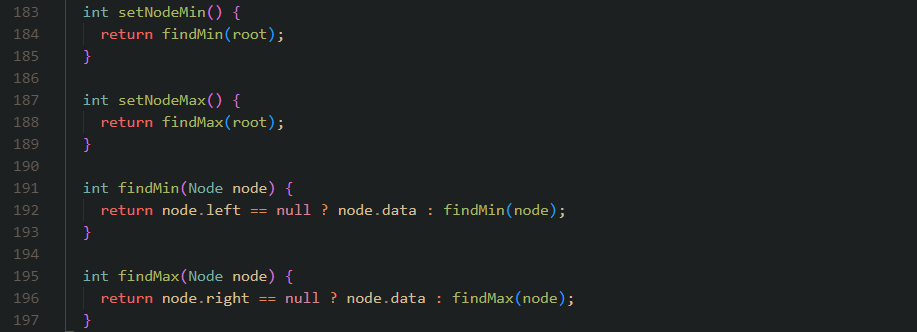
**Jawab :** Untuk menetapkan index terakhir dari tree menjadi 6, dibuat 6 karena data yang valid untuk dimasukkan kedalam tree hanya ada 6.

**Tugas Praktikum**

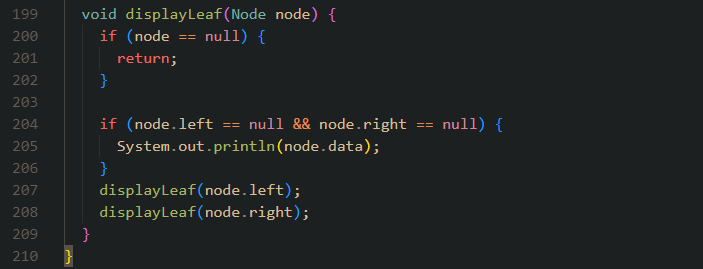
* 1. Buat method di dalam class **BinaryTree** yang akan menambahkan node dengan cara rekursif.

****

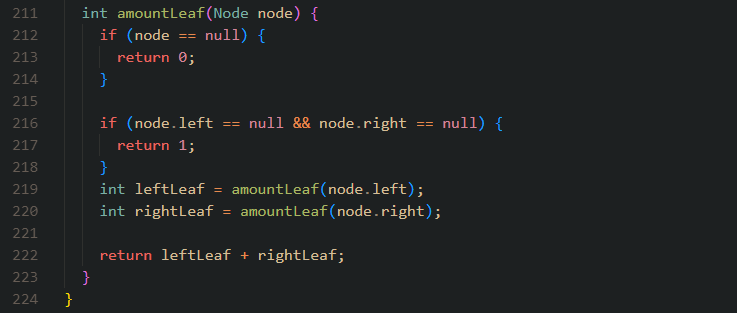
* 1. Buat method di dalam class **BinaryTree** untuk menampilkan nilai paling kecil dan yang paling besar yang ada di dalam tree.

****

* 1. Buat method di dalam class **BinaryTree** untuk menampilkan data yang ada di leaf.

****

* 1. Buat method di dalam class **BinaryTree** untuk menampilkan berapa jumlah leaf yang ada di dalam tree.

****

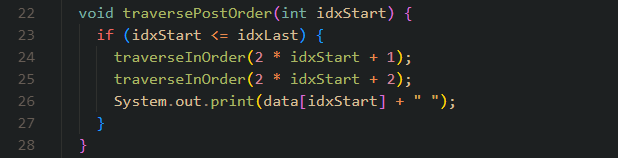
1. Modifikasi class **BinaryTreeArray**, dan tambahkan :

* method **add(int data)** untuk memasukan data ke dalam tree



* method **traversePreOrder()** dan **traversePostOrder()**

Method traversePostOrder()



Method traversePreOrder()

