Nama : Ruth Aulya Silalahi

Kelas : 12 IF1

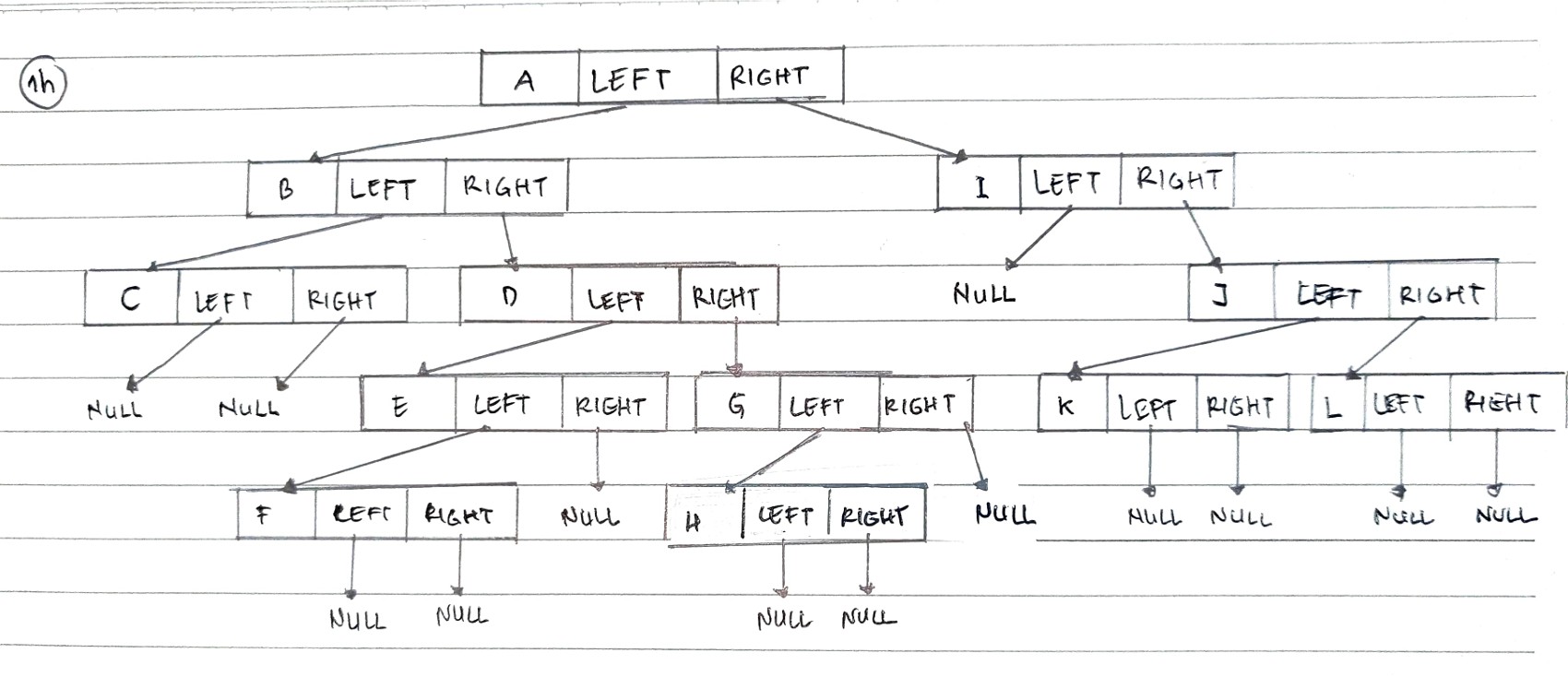
NIM : 11S20018

Topik : Tree, Binary Tree, Binary Search Tree

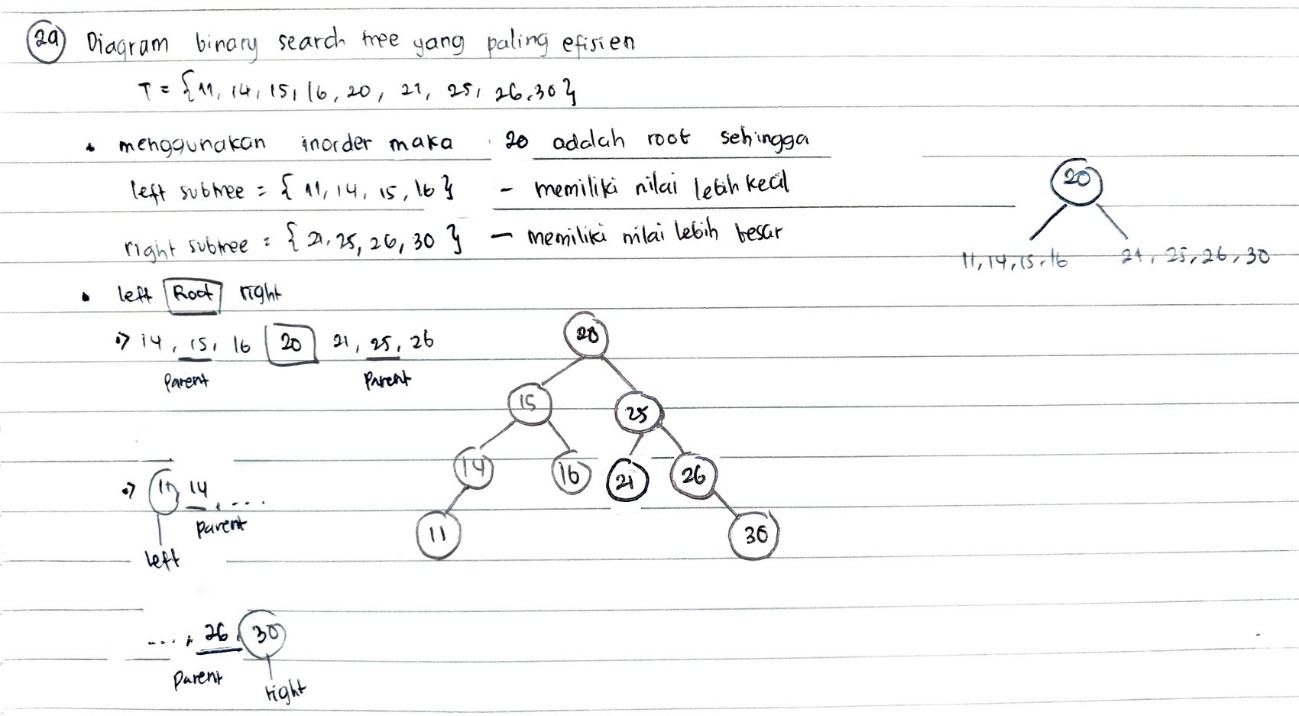
1. **Bagian Pemahaman Konsep**
2. Perhatikan diagram generic tree berikut ini:



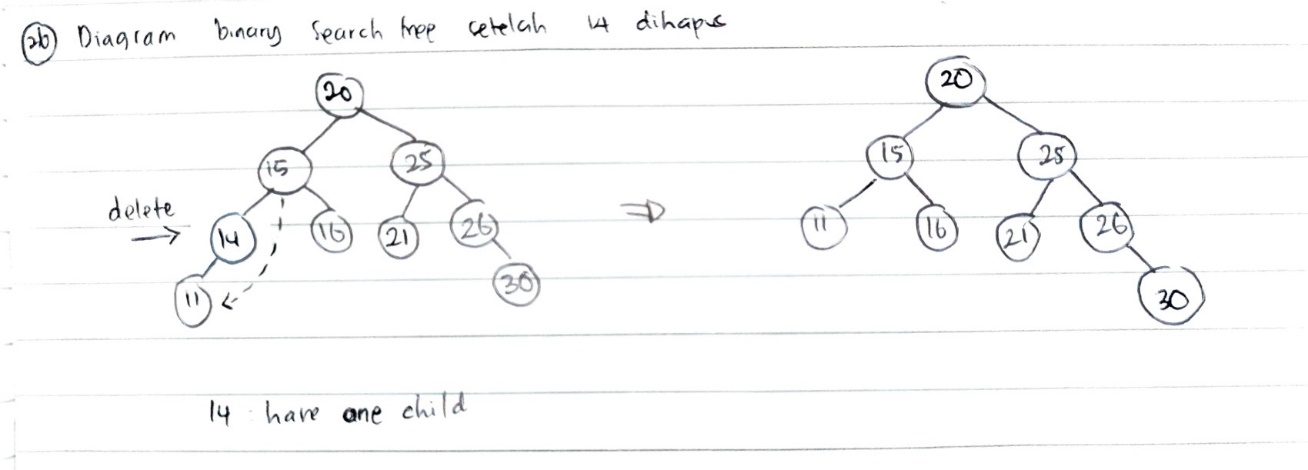
1. Root: Node A
2. Eksternal node (daun): C, F, H, K, L
3. Kedalaman pohon: 4
4. Tinggi pohon: 4
5. Kedalaman node E: 3
6. Descendant dari node B: C, D, E, G, F, H
7. Ancestor dari node J: I, A
8. Skema implementasi generic tree dengan representasi linked list:



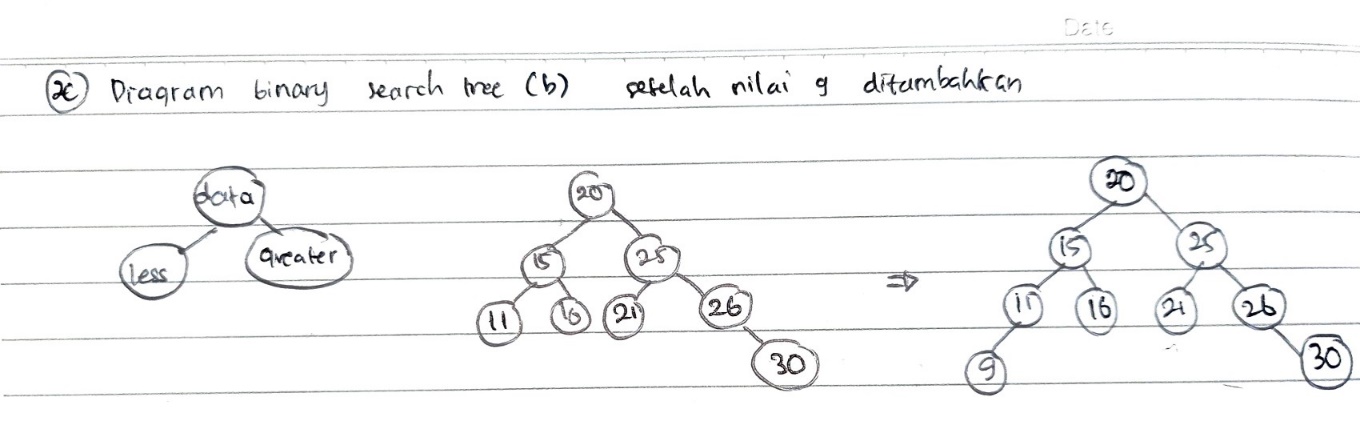
1. Diberikan data sebagai berikut: T = {11, 14, 15, 16, 20, 21, 25, 26, 30}
2. Diagram binary search tree yang paling efisien



1. Diagram binary search tree setelah nilai 14 dihapus



1. Diagram binary search tree setelah nilai 9 ditambahkan



1. **Implementasi**
2. Binary Tree
3. Kode kelas BinaryNode

Link:

[BinaryTree\_11S20018\BinaryNode\_11S20018.java](BinaryTree_11S20018/BinaryNode_11S20018.java)

1. Kode kelas BinaryTree

Link:

[BinaryTree\_11S20018\BinaryTree\_11S20018.java](BinaryTree_11S20018/BinaryTree_11S20018.java)

1. Kode untuk menguji method yang ada

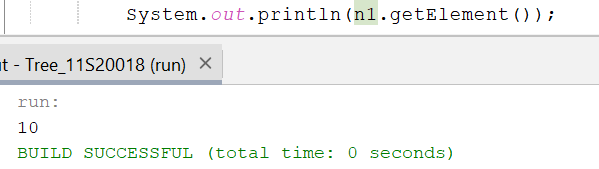
Link:

[BinaryTree\_11S20018\TestBinaryNode\_11S20018.java](BinaryTree_11S20018/TestBinaryNode_11S20018.java)

[BinaryTree\_11S20018\TestBinaryTree\_11S20018.java](BinaryTree_11S20018/TestBinaryTree_11S20018.java)

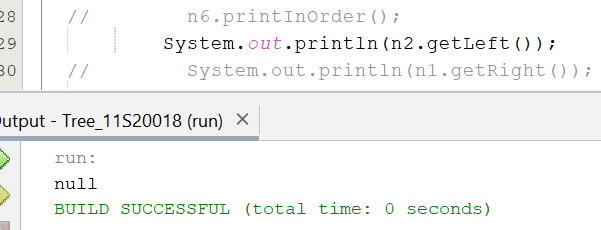
Method getElement: untuk mendapatkan nilai elemen





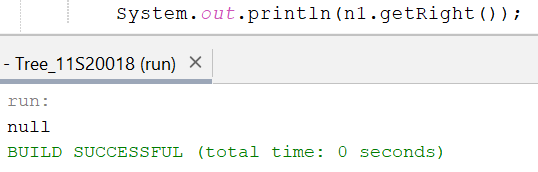
Method getLeft: untuk mendapatkan node sebelah kiri dari sebuah node





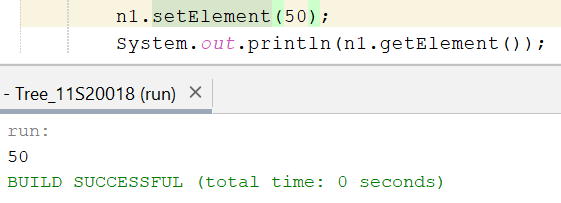
Method getRight: untuk mendapatkan node sebelah kana dari sebuah node



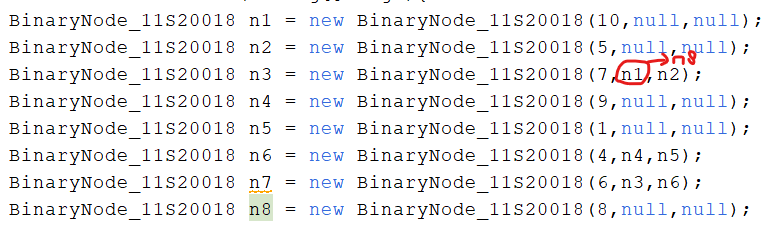


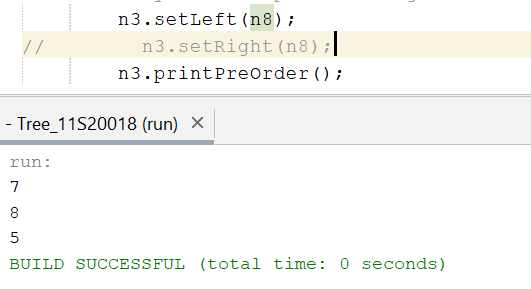
Method setElement: mengatur ulang nilai elemen



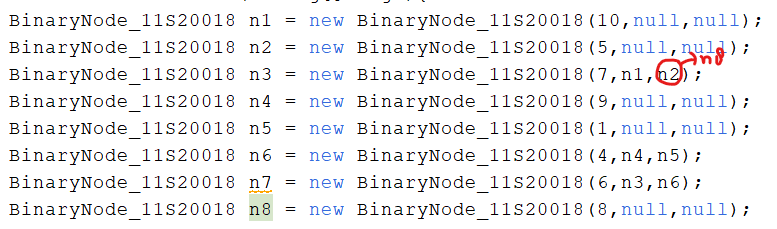


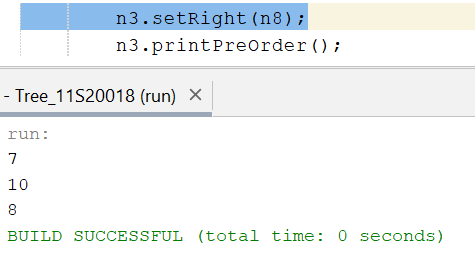
Method setLeft: mengatur ulang nilai sebelah kiri dari sebuah node





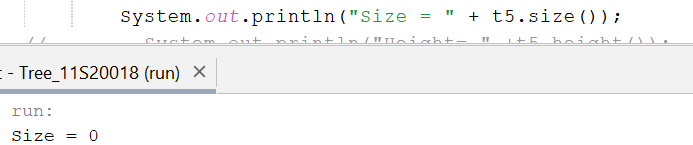
Method setRight: mengatur ulang nilai sebelah kanan dari sebuah node





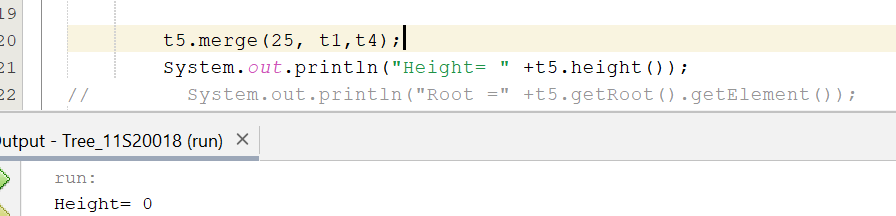
Method size: mengembalikan ukuran dari subtree suatu node



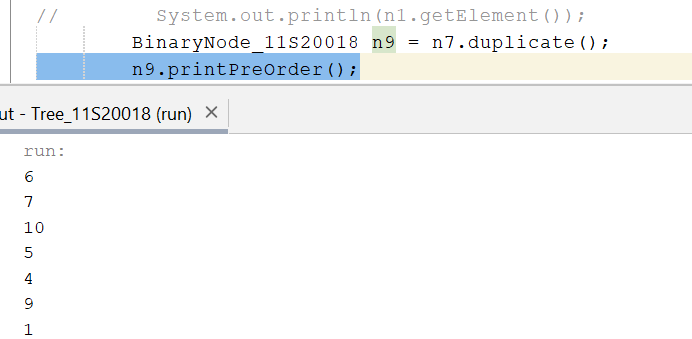


Method height: mengembalikan tinggi dari subtree pada suatu node

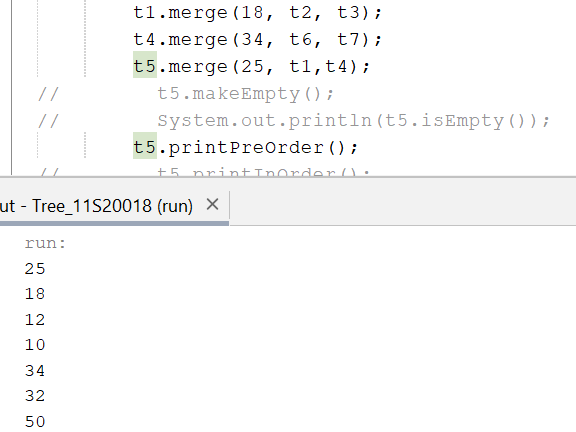




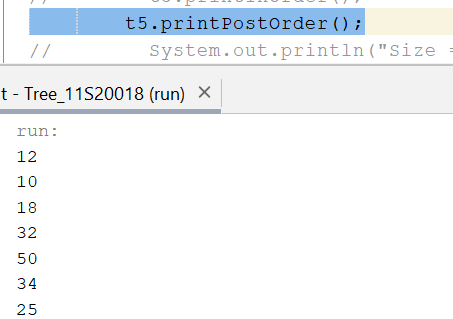
Metode duplicate: mengembalikan duplikat tree



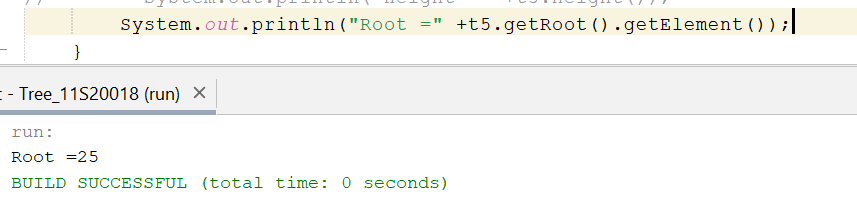
Method printPreOrder: menampilkan tree tranversal secara preOrder



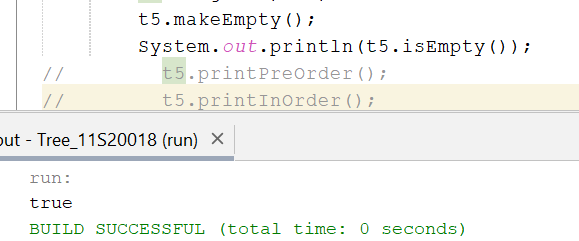
Method printPostOrder: menampilkan tree transversal secara postOrder



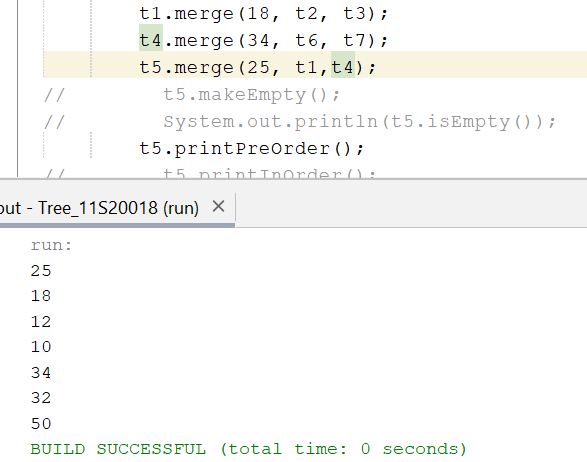
Method printInOrder: menampilkan tree transversal secara inOrder



Method makeEmpty: mengosongkan nilai tree



Method merge: menggabungkan sebuah nilai menjadi nilai node



1. Binary Search Tree
2. Kode kelas BinaryNode

Link:

[BinarySearchTree\_11S20018\BinaryNode\_11S20018.java](BinarySearchTree_11S20018/BinaryNode_11S20018.java)

1. Kode kelas BinarySearchTree

Link:

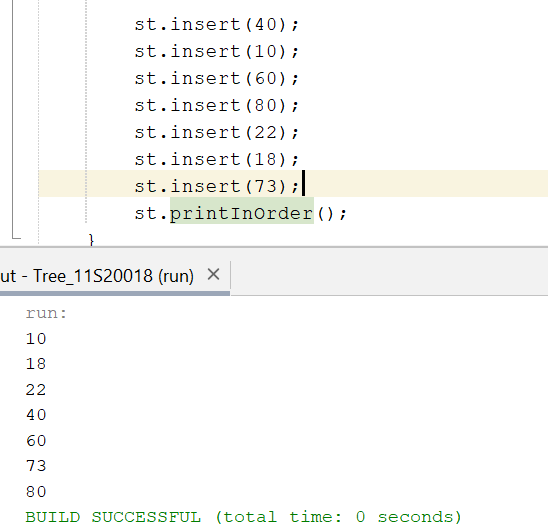
[BinarySearchTree\_11S20018\BinarySearchTree\_11S20018.java](BinarySearchTree_11S20018/BinarySearchTree_11S20018.java)

1. Kode untuk menguji method yang ada

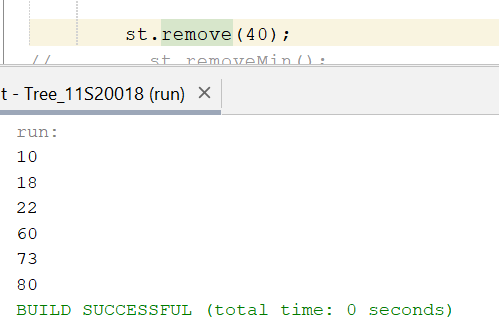
Link:

[BinarySearchTree\_11S20018\TestBinarySearchTree\_11S20018.java](BinarySearchTree_11S20018/TestBinarySearchTree_11S20018.java)

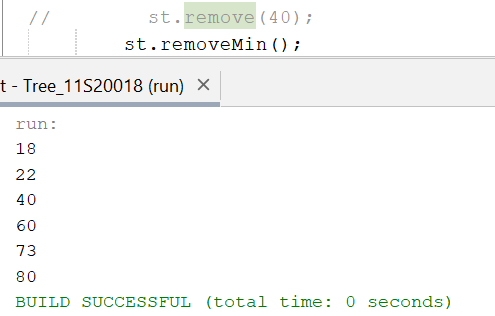
Method insert: menambahkan nilai



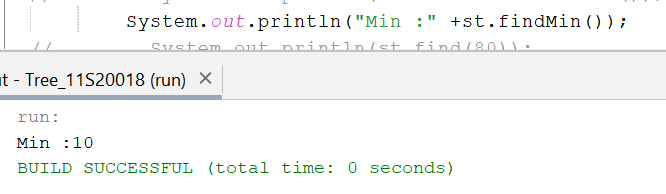
Method remove: menghapus nilai x pada subtree



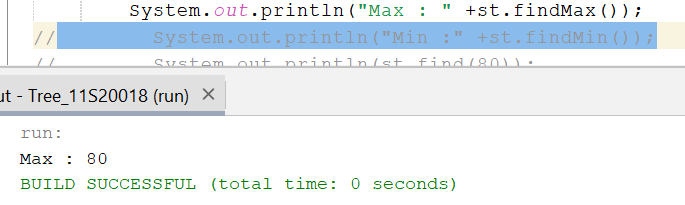
Method removeMin: menghapus nilai minimum pada subtree



Method findMin: mengembalikan item dengan nilai terkecil pada subtree

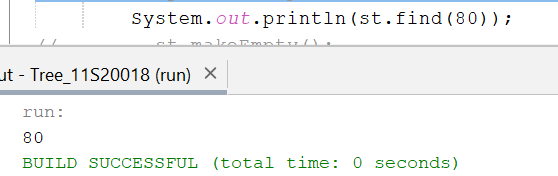


Method findMax: mengembalikan item dengan nilai terbesar pada subtree

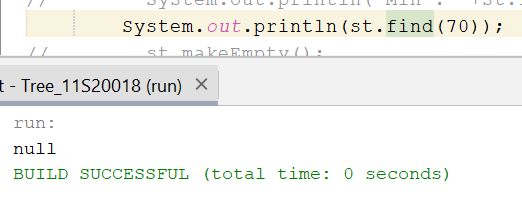


Method find: mengembalikan item yang bernilai sama dengan x yang dicari

Jika ada:



Jika tidak ada:



Method elementAt: mengembalikan nilai elemen t (bersifat private)

Method makeEmpty: mengosongkan nilai tree

Method isEmpty: mengembalikan nilai true jika tree dalam keadaan kosong

