RANGKUMAN BST&SORTING

ILHAM HANAMI

201610370311200

Universitas Muhammadiyah Malang

Fakultas Teknik PRODI Teknik Informatika

Tahun 2017

Binary Search Tree

1. **Binary Search Tree (BST)**

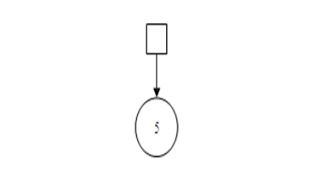
Binary Search Tree adalah tree yang terurut (ordered Binary Tree). Aturan yang harus dipenuhi untuk membangun sebuah BST adalah sebagai berikut:

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

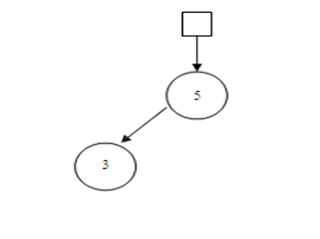
**Pembentukan BST**

Bila diketahui sederetan data 5, 3, 7, 1, 4, 6 maka proses inserting (memasukkan) data tersebut dalam algoritma BST langkah per langkah adalah sebagai berikut.

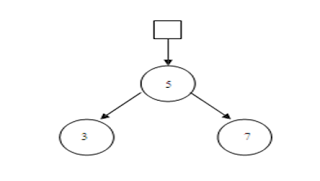
Langkah 1: Pemasukan data 5 sebagai root

[](http://1.bp.blogspot.com/-Z09Mml6zRiw/VUeL3uuWaFI/AAAAAAAAAo0/TGTCJfNoGIw/s1600/3.png)

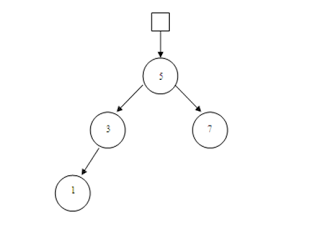
Langkah 2: Pemasukan data 3 disebelah kiri simpul 5 karena 3 < 5.

[](http://4.bp.blogspot.com/-Lp73E1GUryE/VUeMJ3HRzOI/AAAAAAAAAo8/UpJNZ0aGjGo/s1600/4.png)

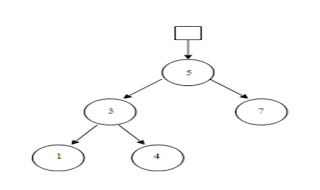
Langkah 3: Pemasukan data 7 disebelah kanan simpul 5 karena 7 > 5.

[](http://1.bp.blogspot.com/-UTTOZ8DT37U/VUeMfpFSHNI/AAAAAAAAApE/8I5FMJrScd0/s1600/5.png)

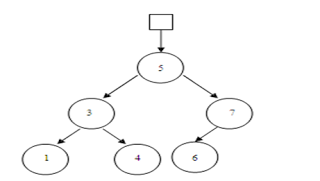
Langkah 4:   Pemasukan data 1. Karena data 1 lebih kecil dari data di root yaitu 5 maka penelusuran dilanjutkan kesebelah kiri root. Kemudian karena disebelah kiri sudah ada daun dengan nilai 3 dan data 1 lebih kecil dari data 3 maka data 1 disisipkan disebelah kiri simpul 3.

[](http://3.bp.blogspot.com/-tGF4oJNEX8w/VUeNTWFOKpI/AAAAAAAAApQ/uqey3gcfH0Q/s1600/6.png)

Langkah 5:   Pemasukan data 4.

[](http://2.bp.blogspot.com/-rYmku04SWhE/VUeN3aeVEQI/AAAAAAAAApY/Q9zx41NQxZI/s1600/7.png)

Langkah 6:   Pemasukan data 6. Karena data 6 lebih besar dari data di root yaitu 5 maka penelusuran dilanjutkan kesebelah kanan root. Kemudian karena disebelah kanan sudah ada simpul dengan nilai 7 dan data 6 lebih kecil dari data 7 maka data 6 disisipkan disebelah kiri simpul 7.

[](http://1.bp.blogspot.com/-DXoldY7z9vQ/VUeOcXDVviI/AAAAAAAAApg/TuqnnkKQiq0/s1600/8.png)

**Insert BST**

Penyisipan sebuah  node baru, didahului dengan operasi pencarian posisi yang sesuai. Dalam hal ini node baru tersebut akan menjadi daun/leaf.

|  |
| --- |
| [https://1.bp.blogspot.com/-hTpkECzSrDI/VJE7OGFg23I/AAAAAAAAAd8/d8ykBM-dmuA/s1600/bst%2Binsert.PNG](http://1.bp.blogspot.com/-hTpkECzSrDI/VJE7OGFg23I/AAAAAAAAAd8/d8ykBM-dmuA/s1600/bst+insert.PNG) |
| Insert BST |

**Delete BST**

Operasi delete memiliki 3 kemungkinan :

-          Delete terhadap node tanpa anak/child (leaf/daun) : node dapat langsung dihapus

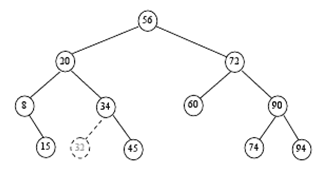
-          Delete terhadap node dengan satu anak/child : maka node anak akan menggantikan posisinya.

-          Delete terhadap node dengan dua anak/child : maka node akan digantikan oleh node paling kiri dari Sub Tree Kanan atau dapat juga digantikan oleh anak paling kanan dari Sub Tree Kiri.

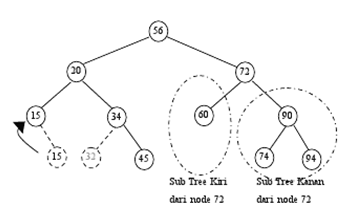
|  |
| --- |
| [https://3.bp.blogspot.com/-bI-1FPrUMuY/VJE4Y2uhiyI/AAAAAAAAAdE/_Xrg7x9RU_E/s1600/bst%2Bdelete.PNG](http://3.bp.blogspot.com/-bI-1FPrUMuY/VJE4Y2uhiyI/AAAAAAAAAdE/_Xrg7x9RU_E/s1600/bst+delete.PNG) |
| Delete BST |

Misalnya ingin dihapus

1. Node (32) : dapat langsung dihapus sehingga akan dihasilkan tree sbb.

[](http://4.bp.blogspot.com/-l2yuN2hovuo/VJE465DT0TI/AAAAAAAAAdQ/SIP9wuIvTKI/s1600/bst+delete1.PNG)

2. Node (8) : node dengan satu child

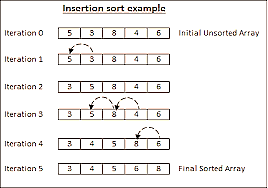
[](http://1.bp.blogspot.com/-SzohIUQVXbo/VJE5harxBzI/AAAAAAAAAdY/LWnWwrbSvkA/s1600/bst+delete2.PNG)

SORTING

1. SORTING
2. INSERTION SORT

Algoritma insertion sortpada dasarnya memilah data yang akan diurutkan menjadi dua bagian, yang belum diurutkan dan yang sudah diurutkan. Elemen pertama diambil dari bagian array yang belum diurutkan dan kemudian diletakkan sesuai posisinya pada bagian lain dari array yang telah diurutkan. Langkah ini dilakukan secara berulang hingga tidak ada lagi elemen yang tersisa pada bagian array yang belum diurutkan.

Contoh :



Gambar 2.1 Insertion Sort

1. SELECTION SORT

Ide utama dari algoritma selection sortadalah memilih elemen dengan nilai paling rendah dan menukar elemen yang terpilih dengan elemen ke-i. Nilai dari idimulai dari 1 ke n, dimana nadalah jumlah total elemen dikurangi 1.

Contoh :

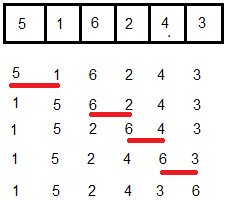


Gambar 2.2 Selection Sort

1. BUBLE SORT

Merupakan algoritma pengurutan paling tua dengan metode pengurutan paling sederhana. Pengurutan yang dilakukan dengan membandingkan masing-masing itemdalam suatu list secara berpasangan, menukar itemjika diperlukan, dan mengulaginya sampai akhir list secara berurutan, sehingga tidak ada lagi item yang dapat ditukar.

Contoh :



Gambar 2.3 Buble Sort