

Implementasi BST(D40)

Buatlah program untuk mengimplementasikan Binary Search Tree. Di dalam implementasi tersebut terdapat operasi-operasi seperti:

- insert x, memasukkan x ke dalam BST
- search x, mencari x di dalam BST dan menampilkan hasil pencariannya
- delete x, menghapus x jika x ada pada BST
- depth, menampilkan tinggi tree saat ini.

Spesifikasi Input

Input berupa interaksi berulang antara user dan program. Input dapat memiliki 5 bentuk, yang masing-masing diawali dengan perintah 'insert', 'search', 'delete', 'depth' dan 'quit'. Sintaks dari tiap bentuk adalah sebagai berikut:

insert <suatu nilai>

Perintah 'insert' digunakan untuk menyimpan nilai ke dalam BST

search <suatu nilai>

Perintah 'search' digunakan untuk mencari nilai dalam BST

delete <suatu nilai>

Perintah 'delete' digunakan untuk menghapus nilai dari BST jika ada

depth

Perintah 'depth' digunakan untuk mengetahui tinggi tree saat ini

quit

Perintah 'quit' digunakan untuk menghentikan eksekusi program

<suatu nilai> adalah bilangan bulat N, dengan $0 \leq N \leq 1000$

Jumlah perintah yang dimasukkan tidak lebih dari 500. Nilai-nilai N yang dimasukkan bersifat acak/uniform dan unik (N tidak pernah diproses lebih dari satu kali untuk perintah **insert**)

Spesifikasi output

Output untuk perintah **insert** adalah dua buah baris dengan baris pertama berisi sebuah string dengan format "Nilai x berhasil dimasukkan", jika x yang merupakan nilai yang akan dimasukkan berhasil dimasukkan. Baris kedua berisi angka-angka yang merupakan isi tree saat ini dan ditampilkan secara inorder. Penulisan pada baris ini diawali dengan tanda "[", dilanjutkan dengan angka-angka pada BST, jika angka-angka tersebut berjumlah lebih dari satu, masing-masing angka dituliskan terpisah oleh tanda koma dan spasi, dan baris ii diakhiri oleh tanda "]".

Output untuk perintah **search** adalah sebuah string dengan format "Nilai x ditemukan", jika data ditemukan dengan x adalah nilai yang dicari, atau "Nilai x tidak ditemukan" jika nilai yang dicari tidak ada.

Output untuk perintah **delete** adalah dua buah baris dengan baris pertama berisi sebuah string dengan format "Nilai x telah dihapus", jika x yang merupakan nilai yang akan dihapus ada pada tree. Namun jika nilai tersebut ternyata tidak ada maka format yang ditampilkan "Nilai x tidak ditemukan". Baris kedua berisi angka-angka yang merupakan isi tree setelah penghapusan terjadi/tidak terjadi dan ditampilkan secara inorder dengan penulisan seperti pada perintah **insert**.

Output untuk perintah **depth** adalah sebuah string dengan format “Tinggi tree saat ini adalah x”, dengan x adalah tinggi tree.

Contoh Input

```
insert 3
insert 5
insert 4
insert 1
insert 7
insert 8
insert 2
insert 9
insert 6
find 3
delete 5
find 5
delete 5
depth
quit
```

Contoh Output

```
Nilai 3 berhasil dimasukkan
[3]
Nilai 5 berhasil dimasukkan
[3, 5]
Nilai 4 berhasil dimasukkan
[3, 4, 5]
Nilai 1 berhasil dimasukkan
[1, 3, 4, 5]
Nilai 7 berhasil dimasukkan
[1, 3, 4, 5, 7]
Nilai 8 berhasil dimasukkan
[1, 3, 4, 5, 7, 8]
Nilai 2 berhasil dimasukkan
[1, 2, 3, 4, 5, 7, 8]
Nilai 9 berhasil dimasukkan
[1, 2, 3, 4, 5, 7, 8, 9]
Nilai 6 berhasil dimasukkan
[1, 2, 3, 4, 5, 6, 7, 8, 9]
Nilai 5 telah dihapus
[1, 2, 3, 4, 6, 7, 8, 9]
Nilai 5 tidak ditemukan
[1, 2, 3, 4, 6, 7, 8, 9]
Tinggi tree saat ini adalah 5
```