

Laporan Praktikum

Tree



Disusun oleh:

Tomi Martino Affandi

2341720245

D-IV Teknik Informatika - 1E

Jl. Soekarno Hatta No.9, Jatimulyo, Kec. Lowokwaru, Kota Malang Jawa Timur 65141

Phone : (0341)404424, 404425

E-mail : Polinema.ac.id

11.1 Tujuan Praktikum

Setelah melakukan praktikum ini, mahasiswa mampu:

1. memahami model Tree khususnya Binary Tree
2. membuat dan mendeklarasikan struktur algoritma Binary Tree.
3. menerapkan dan mengimplementasikan algoritma Binary Tree dalam kasus Binary Search Tree

11.2.1 Percobaan 1

Code:

<https://github.com/TomiMartinoAffandi/Algoritma---Struktur-Data/commit/3e6669c471d61d9c02f03f153bba3aa7c0f3509b>

Output:

```
PreOrder Traversal: 6 4 3 5 8 7 9 10 15
InOrder Traversal: 3 4 5 6 7 8 9 10 15
PostOrder Traversal: 3 5 4 7 15 10 9 8 6
Find Node: true
Delete Node: 8
```

```
PreOrder Traversal: 6 4 3 5 9 7 10 15
```

```
PS D:\TUGAS KULIAH\SEMESTER 2\Algoritma & Struktur Data> █
```

Pertanyaan

1. Karena dalam Binary Search Tree(BST) node dipilah menjadi 2 bagian yaitu node sebelah kiri akan selalu lebih kecil dari node sebelah kanan sehingga memudahkan dalam pencarian
2. Untuk memisahkan node, yaitu semua node yang kecil berada di sebelah kiri dan sisanya berada di kanan
3.
 - a) Seperti analogi pada pohon root digunakan untuk percabangan dan setiap root memiliki node(isi)
 - b) Nilai root = null
4. Pengecekan apakah root kosong atau tidak jika kosong maka data akan ditambah di paling awal
5. Jika data yang ingin ditambahkan lebih kecil dari data sebelumnya maka akan dilakukan pengecekan lagi, jika terdapat data pada root bagian kiri maka pointer akan dilanjutkan ke root kirinya lagi jika ditemukan null maka data baru akan ditambahkan

11.3.1 Tahapan Percobaan

Code:

<https://github.com/TomiMartinoAffandi/Algoritma---Struktur-Data/commit/53e68060acc2222d51717e966718334310dc2ee7>

Output:

```
InOrder Traversal: 3 4 5 6 7 8 9
```

```
PS D:\TUGAS KULIAH\SEMESTER 2\Algoritma & Struktur Data>
```

Pertanyaan:

1. Data digunakan untuk menyimpan nilai data kedalam array data, idxlast untuk ukuran yang digunakan ketika tranversal
2. Untuk mengisi nilai pada atribut data dan menentukan menentukan panjang idxlist
3. Untuk mencetak data dari kiri(terkecil) ke root lalu ke kanan(terbesar)
4. Left child 4 dan right child 5
5. Nilai yang digunakan untuk tranversal(karena dimulai dari index 0 maka idxlast = 6)