MENCARI AHLI STRUKTUR DATA (MCS)

**Deskripsi Soal**

Bejo diminta untuk membuat sistem yang dapat membangun struktur data AVL Tree dan menghitung jumlah antara parent sebuah node dengan siblingnya (Sibling merupakan dua node dari parent yang sama).

**Ide Penyelesaian**

* pada soal ini saya membuat dua function diluar main yang inti dari function nya adalah untuk 1.) Mencari parent dan 2.) mencari sibling. Kemudan akan dicari parent dari value yg dicari, value parent tsb akan dipakai untuk mencari siblingnya parent menggunakan fungsi kedua. Lalu, parent dan siblignya dari parent tsb akan dijumlahkan dan dikeluarkan sbg output.

**Tahapan Penyelesain (Secara Umum / IDE Awal)**

1. Membuat scanf yg dimasukin ke while untuk menerima inputan
2. Membuat fungsi ‘findParent’ diluar main
3. Membuat fungsi ‘findSibling’ diluar main
4. Masukan value parent yang dicari kedalam suatu variabel
5. Cari value sibling dari parent
6. Jumlahkan value parent dan sibling kemudian print

**Link Code**

<https://ideone.com/cvf8qK>

**Penjelasan Kode (Secara Detail / Setelah Di implementasi)**

* Line 230: Membuat fungsi rekursif findparent
* Line 231: base condition untuk fungsi
* Line 233: if untuk ngecek root yg skrg kita cek itu udah sama dengan val atau belum. Jika sama, maka value variabel parent akan di assign kepada alamat yang ditunjuk pointer hasil
* Line 235: Jika tidak sama maka akan masuk ke else-nya, yang akan memanggil findparent dengan parameter, root ke kiri, value yang dicari, parent tetap pada value node sebelumnya, pointer hasil
* Line 237: Memanggil findparent dengan parameter, root ke kanan, value yang dicari, parent tetap pada value node sebelumnya, pointer hasil
* Line 240: Membuat fungsi primary untuk fungsi findparent
* Line 243: Membuat fungsi rekursif findsibling
* Line 244: base condition untuk fungsi findsibling
* Line 246: if untuk mengecek apakah val sudah sama atau belum dengan yang kita cari
* Line 247: Jika line 246 benar, maka akan masuk ke if yang mengecek jika value pada parent sama dengan value pada val
* Line 248: Jika line pada 247 benar maka akan masuk ke if yang akan mengecek child kanan dari parent ada atau tidak. Jika iya maka value dari child kanan node parent di assign ke alamat yang ditunjuk pointer sibling
* Line 250: Jika line 248 benar, maka akan mengassign nilai 0 pada alamat yang ditunjuk oleh pointer sibling
* Line 253: Jika line 247 salah, maka akan masuk ke else if yang akan mengecek jika value parent lebih kecil dari value val atau tidak
* Line 254: Jika line 253 benar, maka akan masuk if yang mengecek kalau child node kiri dari parent ada atau tidak, jika iya maka value child node kiri dari parent akan diassign ke alamat yang ditunjuk pointer sibling
* Line 256: Jika line 254 false maka akan masuk ke else yang akan mengassign 0 ke alamat yang ditunjuk pointer sibling
* Line 259: Jika if line 247 false, kemudian else if pada line 253 juga false, maka akan masuk ke else yang akan mengassign 0 ke alamat yang ditunjuk oleh pointer sibling
* Line 262: Jika if pada line 246 adalah false maka akan masuk ke else, yang akan menjalankan line 263 dan 264
* Line 263: Memanggil fungsi findsbibling dengan parameter root ke kiri, value yang dicari, root(node sebelumnya) pada parameter parent, dan pointer sibling
* Line 264: Memanggil fungsi findsbibling dengan parameter root ke kanan, value yang dicari, root(node sebelumnya) pada parameter parent, dan pointer sibling
* Line 268: Membuat fungsi primary untuk fungsi findsibling
* Line 272: int main
* Line 273: deklarasi variable integer n = untuk insert brp banyak data yg dijakdikan node; x = value/angka yg akan dimasukan ke node; t= berapa banyak node yg akan dicek; b= value yg akan di cek
* Line 277 & 278: scanf untuk x dan t
* Line 280: while untuk masukin input angka kedalam node sesuai dengan inputan n
* Line 281: scanf untuk insert x ke node
* Line 282: untuk insert x ke dalam tree
* Line 286: while untuk pengulangan node yg dicek parent nya. Pengulangan nya sesuai dgn inputan t
* Line 287 288: deklarasi variable integer data, parent, sibling. Di define jadi 0 karena biar default
* Line 289: scanf untuk b (nilai yg ingin dicari)
* Line 290 291: memanggil fungsi findParentFin dan findSiblingFin
* Line 292: menghitung total value parent + value siblingnya parent (sekalgus pendeklarasian variable int total)
* Line 293: printf untuk hasil total line diatas alias outputnya

**Referensi**

* Ranto Sitorus (5025221228)
* Mikhael Abie Saputra (5025221113)