Muhammad Raihan
1806141366

Penjelasan Ide TP-3

Pengambilan Input

Pada TP 3 ini saya menggunakan template yang sama yang digunakan untuk mengambil input dan mengeluarkan output pada TP 2 dengan menambah beberapa class dan method berbeda.

Struktur Data yang Digunakan

Saya menggunakan struktur data tree dengan HashMap didalam tree tersebut. Saya membuat 3 class tambahan yaitu “Wilayah”, yang memiliki atribut id dan jumlah suara 1 dan 2 di wilayah tersebut. Kedua adalah class Node dengan penggunaan generic, tetapi saya hanya menggunakan node tersebut dengan isi Wilayah. Node memiliki atribut data, Node yang menjadi parent dari Node tersebut, serta ArrayList yang berisi Node yang menjadi child dari Node tersebut. Ketiga adalah class Tree, yang memiliki atribut Node sebagai root dari tree tersebut, dan HashMap dengan key berupa String dan value berupa Node untuk mencatat dan mengambil Node yang ada dalam tree tersebut.

Flow Program

- Pengambilan Input
  Saya mengambil 2 String pertama sebagai donat1 dan donat2, untuk membedakan mengambil suara vote1 atau 2 kemudian. Saya mengambil integer selanjutnya sebagai banyak sub-wilayah yang akan dibuat. Ambil baris pertama dari wilayah untuk membuat wilayah nasional, lalu jadikan wilayah sebagai root, dan masukkan wilayah provinsi sebagai child dari nasional. Lakukan loop sebanyak banyak sub-wilayah-1 untuk memasukkan sub-wilayah yang lebih kecil ke dalam tree. Setelah itu, saya mengambil integer selanjutnya dan melakukan loop sebanyak integer tersebut untuk melakukan perintah yang di input
  Kompleksitas : O(banyakWilayah)

- Method tambahSuara()
  Jika instruksi yang dimasukkan adalah “TAMBAH”, maka saya mengambil 1 String selanjutnya sebagai nama wilayah yang akan ditambah, dan 2 long selanjutnya sebagai angka yang ditambahkan pada masing-masing suara di wilayah tersebut. Saya mengambil Node dengan Wilayah sesuai dengan nama wilayah yang dimasukkan sebelumnya di HashMap pada Tree, lalu saya menjalankan method tambahSuara dengan 3 parameter, yaitu Node, dan 2 buah long. Dalam method tambahSuara, saya menambahkan masing-masing index long array dalam Node tersebut sebanyak angka yang dimasukkan sebelumnya. Kemudian jika Node tersebut mempunyai parent, maka dilakukan method tambahSuara kepada parent dari Node tersebut.
  Kompleksitas : O(log(jumlah Node)).

Muhammad Raihan
1806141366

- Method kurangSuara()
  Method kurangSuara sama seperti method tambahSuara, tetapi setiap index dari long array pada Node tersebut tidak ditambah melainkan dikurang dengan long yang di-input user sebelumnya. Jika Node tersebut masih mempunyai parent, maka akan dilakukan kurangSuara pada parent dari Node tersebut.
  Kompleksitas : O(log(jumlah Node)).

- Method cekSuara()
  Dalam method cekSuara, saya mengambil 1 buah String sebagai id dari wilayah yang ingin dicetak jumlah suaranya, lalu saya mengambil Node dengan id tersebut di HashMap pada Tree. Kemudian saya mencetak suara 1 dan 2 yang ada pada array Long di Wilayah tersebut.
  Kompleksitas : O(1)

- Method wilayahMenang()
  Sebelum menjalankan method wilayah, saya mengambil String input, jika input tersebut sama dengan donat pertama, maka memasukkan parameter 1, sebaliknya akan 0. Dalam method wilayahMenang, saya melakukan loop pada semua value di HashMap, lalu membandingkan elemen index ke parameter dalam long array setiap Wilayah dalam Node, jika index ke-parameter lebih besar dari index lainnya, maka counter akan betrambah.
  Kompleksitas : O(jumlah Wilayah).

- Method cekSuaraProvinsi()
  Jika instruksi yang dimasukkan adalah “CEK_SUARA_PROVINSI”, maka saya melakukan loop pada ArrayList child pada root, lalu mencetak id pada Node tersebut dan mencetak suara 1 dan 2.
  Kompleksitas : O(Jumlah Provinsi)

- Method wilayahMinimal()
  Dalam method wilayahMinimal, saya melakukan loop untuk memeriksa semua Node dalam HashMap di Tree, kemudian saya menghitung persentase donat sesuai input, jika lebih besar dari minimal persentase yang dibutuhkan maka counter bertambah.
  Kompleksitas : O(banyak wilayah).

- Method wilayahSelisih()
  Untuk menghitung selisih, saya melakukan loop untuk semua Node dalam Tree, lalu menghitung selisih kedua suara di dalam Node tersebut, jika selisihnya lebih besar atau sama dengan selisih minimal yang di input, maka counter bertambah.
  Kompleksitas : O(banyak Wilayah).

Kompleksitas Total : O(banyak Wilayah).