

Nama : Ali Irsyaad Nursyaban  
NPM : 1806141132  
Kelas : SDA C

## PENJELASAN IDE TP-3 SDA

Pada TP3 PemiluDonatRaya kali ini saya membuat 4 class (selain public class untuk main program) antara lain Pemilu, Wilayah, Node, dan Tree. Untuk perintah terdapat TAMBAH, ANULIR, CEK\_SUARA, CEK\_SUARA\_PROVINSI, WILAYAH\_MENANG, WILAYAH\_MINIMAL, dan WILAYAH\_SELISIH. Saya akan jelaskan satu per satu.

### 1. Class Pemilu

Class ini berfungsi sebagai tempat menyimpan root dari wilayah yaitu wilayah nasional, dan method-method untuk setiap perintah. Dalam class ini juga menyimpan nama kandidat1 dan kandidat2, HashMap setiap wilayah dan ArrayList wilayah provinsi (nantinya digunakan untuk perintah cek suara provinsi). Alasan saya membuat class ini adalah jika terdapat pemilu untuk negara lain dan dua kandidat yang berbeda maka dapat dibuat object baru dalam class ini

### 2. Class Wilayah

Berfungsi untuk menyimpan informasi dari setiap wilayah: nama wilayah, jumlah suara kandidat satu dan dua, dan wilayah atasnya (parent/super wilayah). Menyimpan super wilayah berfungsi pada perintah tambah dan anulir suara yang hanya dilakukan pada wilayah kelurahan sehingga suara dapat diupdate sampai nasional dengan mengakses wilayah parent (super wilayah).

### 3. Class Node

Merupakan Node biasa dalam AVLTree dengan sedikit tambahan atribut yaitu count untuk menyimpan banyaknya wilayah dengan jumlah selisih/persen yang sama, Sehingga jumlah count dalam setiap Node dalam AVLTree tersebut merupakan jumlah wilayah. Dan tambahan atribut weightLeft dan weightRight untuk menghitung jumlah total count di pada leftNodenya dan rightNodenya.

### 4. Class Tree

Berfungsi sebagai AVLTree biasa yang menyimpan root Node terdapat method insert, remove dan method-method untuk rebalancing : singlerotation dan doublerotation. Terdapat juga method findMin yang digunakan pada method remove yang menggunakan Successor Inorder.

### 5. Tree Selisih dan Persen

Saya menginisiasi 2 object pada Tree yaitu object selisih dan persen. Tree selisih/persen berguna untuk menyimpan selisih/persentase suara kandidat 1 pada setiap wilayah dan mengurutkannya dalam aturan AVLTree with count. Pada saat awal sebelum masuk ke perintah saya memasukkan/insert nilai 0 pada Tree selisih dan nilai 50 pada Tree persen

di setiap menginisiasi object wilayah : `selisih.insert(00);`  
`persen.insert(500);` . Tree selisih dan persen akan diupdate terus menerus pada setiap perintah tambah dan anulir suara dengan cara meremove nilai sebelum ditambah/dianulir dan menginsert nilai sesudahnya. Tree selisih digunakan untuk perintah wilayah\_selisih, dan Tree persen digunakan untuk perintah wilayah\_menang dan wilayah\_minimal.

6. Perintah TAMBAH dan ANULIR suara

Perintah ini selalu menunjuk pada kelurahan sehingga pasti super wilayah atasnya juga berubah suaranya maka saya looping dalam method tambah di class Wilayah wilayah kelurahan sampai wilayah nasionalnya dan menambahkan/menganulir suaranya. Dan saya looping kembali wilayah kelurahan sampai nasional untuk mengupdate AVLTree selisih dan persennya.

7. Perintah WILAYAH\_SELISIH

Pada perintah ini kita mempergunakan Tree selisih yang telah dibuat sebelumnya. Kita tinggal melakukan perbandingan untuk nilai yang dimasukkan dengan value setiap node yang dilewati. Jika value node tersebut lebih besar sama dengan nilai yang dimasukkan maka kita akan menambahkan countnya dan weightrightnya dan melakukan rekursif untuk node leftnya. Sehingga kita mendapatkan hasil jumlah wilayah dengan jumlah minimal selisihnya nilai yang dimasukkan.

8. Perintah WILAYAH\_MENANG dan WILAYAH\_MINIMAL

Pada prinsipnya wilayah menang adalah wilayah minimal yang persentasenya 50% + 1. Algoritma pencarian jumlah wilayahnya hampir sama dengan wilayah selisih hanya saja saya perlu handle untuk kandidat 2 nya karena kita hanya memiliki tree persen kandidat 1. Jadi saya membuat method jumlah\_maksimal yang diberikan argument 100 dikurangi persen masukkan jika yang dicari adalah kandidat 2. Sehingga kita mempergunakan 1 Tree saja.

## Kompleksitas

Untuk menerima input :

$$O(A) + O((B-1) * (C+D+E))$$

Keterangan:

A = banyak wilayah provinsi

B = banyak wilayah kecuali kelurahan

C = banyak kota

D = banyak kecamatan

E = banyak kelurahan

Untuk perintah :

$$O(F * \log G)$$

Keterangan:

F = banyak perintah

G = banyak wilayah

Untuk keseluruhan :

$$O(N*n) + O(M \log N)$$

Keterangan:

N = banyak wilayah

n = banyak subWilayah

M = banyak perintah

\*  $\log N$  adalah kompleksitas untuk insert, remove, dan mencari wilayah selisih, minimal, dan menang

\*  $O(N*n)$  adalah perkiraan untuk kompleksitas menerima input

Sumber :

Untuk class Node dan class Tree saya mendapatkan idenya dari : <https://www.geeksforgeeks.org/avl-with-duplicate-keys/>