Gagas Praharsa Bahar

13520016

1. Collection yang tepat untuk chart lagu di radio adalah sebuah ArrayList. Mengapa bukan Queue atau Stack? Karena radio harus dapat mengakses urutan manapun dari lagu dan juga dapat mencari posisi lagu tersebut di chart minggu ini. Selain itu, karena ArrayList adalah sebuah array kontigu, maka proses searching akan lebih cepat. Hashmap sebenarnya lebih cepat, namun apabila menggunakan HashMap, kita akan tetap search berdasarkan value yang kompleksitasnya tetaplah besar. Untuk implementasi searching, dapat digunakan arrayList.indexOf().

|  |
| --- |
| public static void main(String[] args) {      ArrayList<String> listLagu = new ArrayList<String>();      listLagu.add("Hello");      listLagu.add("World");      int posisi = listLagu.indexOf("World") + 1;      System.out.println("Posisi lagu World: " + posisi);    } |

1. History order makanan dapat diimplementasikan menggunakan ArrayDeque. History merupakan sesuatu yang dapat digambarkan dengan Queue (paling belakang akan dihapus pada waktu tertentu, dan penambahan akan dimulai dari depan, waktu terbaru). Item history yang terdiri atas nama resto, list makanan, dan total harga dapat diimplementasikan menggunakan sebuah kelas. List makanan tersebut dapat dijadikan sebuah array of string biasa, apabila tidak ada kelas makanan lebih lanjut. Akan ada fungsi getItemByResto yang memfilter item berdasarkan resto dengan menggunakan fungsi Stream API filter(). Untuk menghitung jumlah total harga, dengan mempertimbangkan efisiensi memori, akan selalu dicatat ketika makanan dimasukkan kedalam list makanan.

|  |
| --- |
| 1. public class History { 2. private String namaResto; 3. private String[] listMakanan; 4. private int totalHarga; 6. public History(){ 7. this.namaResto = ""; 8. this.listMakanan = new String[5]; 9. this.totalHarga = 0; 10. } 11. public void getItemByResto(String namaResto){ 12. // Mencari item berdasarkan resto 13. } 14. public int getTotalHarga(){ 15. return this.totalHarga; 16. } 17. } 18. public class DaftarHistory { 19. private ArrayDeque<History> daftarHistory; 20. public DaftarHistory(){ 21. this.daftarHistory = new ArrayDeque<History>(); 22. } 23. public void addHistory(History history){ 24. this.daftarHistory.push(history); 25. } 26. public void removeHistory(History history){ 27. this.daftarHistory.remove(history); 28. } 29. public void printHistory(){ 30. for(History history : this.daftarHistory){ 31. history.printHistory(); 32. } 33. } 34. } |