

```

1  /*
2  * Nama : Muhammad Arif Irfan
3  * Nim : 202210370311030
4  * Kelas : Struktur Data J
5  */
6
7  class MyHash1 {
8
9      int maxSize; // ukuran maksimum dari array hash
10     int[] array; // array untuk menyimpan data dalam hash table
11
12     /**
13      * @param maxSize jumlah ukuran atau index untuk array
14      */
15     public MyHash1(int maxSize) { // ! maxSize = 10
16         this.maxSize = maxSize; // menginisialisasi ukuran maksimum array hash
17         this.array = new int[maxSize]; // inisialisasi array dengan ukuran maksimum
18     }
19
20     /**
21      * @param data angka
22      * @return hasil modulus dari data (integer) dan maxSize
23      */
24     public int hashFunction(int data) { // todo : menghitung modulus
25         return data % maxSize; // mengembalikan hasil dari modulus data dan maxSize
26     }
27
28     /**
29      * @param newData angka yang akan di modulus dengan maxSize
30      * @implNote newData akan masuk ke dalam method hashFunction dan akan mengembalikan nilai modulus
31      * nilai modulus akan menjadi nilai index untuk newData yang akan di simpan di dalam array
32      */
33     public void insert(int newData) { // todo : menambahkan data
34         int index = hashFunction(newData); // hasil return dari method hashFunction di inisialisasikan menjadi index
35         array[index] = newData; // hasil dari method hashFunction akan di jadikan index untuk menyimpan newData pada array
36     }
37
38     /**
39      * @param data angka
40      * @return hasil result
41      * @implNote data akan masuk ke dalam method hashFunction lalu hasil dari hashFunction
42      * akan di inisialisasikan menjadi index agar index tersebut di jadikan 0 untuk menghapus data di dalamnya
43      */
44     public int remove(int data) { // todo : menghapus data
45         int result = 0; // mengatur result menjadi 0
46         int index = hashFunction(data); // hasil return dari method hashFunction di inisialisasikan menjadi index
47         result = array[index]; // result adalah array pada index yang dihasilkan hashFunction
48         array[index] = 0; // mengatur kembali array pada index menjadi 0 atau menghapus data pada index tersebut
49         return result; // mengembalikan hasil result
50     }
51
52     /**
53      * @param data angka yang dicari
54      * @return mengembalikan hasil dari data yang di cari
55      * @implNote data yang akan di cari akan di hitung nilai hashnya
56      * jika outputnya sesuai dengan indexnya maka result akan menjadi array pada index tersebut.
57      * jika outputnya 0 maka data tidak tersedia
58      */
59     public int find(int data) { // todo : mencari data
60         int result = 0; // mengatur result menjadi 0
61         int index = hashFunction(data); // hasil return dari method hashFunction
62         result = array[index]; // result adalah array pada index
63         return result; // mengembalikan hasil result
64     }
65
66     /**
67      * @implNote method ini akan melakukan perulangan dari 0 hingga jumlah maxSize
68      * jika array pada index "i" tidak sama dengan 0 maka akan mengouput nilai data pada array itu sendiri
69      */
70     public void display() { // todo : menampilkan data
71         for(int i=0; i<maxSize; i++){ // melakukan perulangan dari 0 hingga maxSize
72             if(array[i] != 0){ // jika kondisi pada array index i != 0
73                 System.out.print(array[i]+" "); // maka akan mengouput data dari array index i itu sendiri
74             }
75         }
76     }
77
78     public class DasarHash1 {
79         public static void main(String[] args) {
80             MyHash1 my = new MyHash1(10); // jumlah index dan akan dijadikan sebagai modulus
81
82             my.insert(10); // memasukan angka 10
83             my.insert(3); // memasukan angka 3
84             my.insert(5); // memasukan angka 5
85
86             my.remove(3); // menghapus angka 3
87
88             System.out.println(my.find(5)); // mencari angka 5
89             System.out.println(my.find(1)); // mencari angka 1
90
91             my.display(); // menampilkan semua data
92         }
93     }
94 }

```