

11520018

Ruth Aulya Silalahi

1) Hitunglah kompleksitas waktu dari operasi pencarian Hash table dengan kondisi dapat menampung sejumlah  $N$  elemen.

Jawab:

Kompleksitas waktu menampung sejumlah  $N: O(1)$ 

2) Insert data berikut ke dalam hash table menggunakan fungsi hash =  $K \bmod \text{TableSize}$  dan strategi penyelesaian collisions menggunakan quadratic probing

insert: 13, 44, 103, 113, 2

		113	13	44		2	103		
0	1	2	3	4	5	6	7	8	9

Jawab:

$$h = K \bmod \text{table size}$$

→ insert 13

$$h = 13 \bmod 10$$

 $h = 3$ , diletakkan pada indeks 3

→ insert 44

$$h = 44 \bmod 10$$

 $h = 4$ , diletakkan pada indeks 4

→ insert 103

$$h = 103 \bmod 10$$

 $h = 3$ , karena indeks 3 telah terisi maka lakukan 1<sup>st</sup> probe

$$h = (h(103) + 1) \bmod 10$$

 $h = 4$ , karena indeks 4 telah terisi maka lakukan 2<sup>nd</sup> probe

$$h = (h(103) + 4) \bmod 10$$

 $h = 7$ , diletakkan pada indeks 7

→ insert 113

$$h = 113 \bmod 10$$

 $h = 3$ , karena indeks 3 telah terisi maka lakukan 1<sup>st</sup> probe

$$h = (h(3) + 1) \bmod 10$$

 $h = 4$ , karena indeks 4 telah terisi maka lakukan 2<sup>nd</sup> probe

$$h = (h(103) + 4) \bmod 10$$

 $h = 7$ , karena indeks 7 telah terisi maka lakukan 3<sup>rd</sup> probe

$$h = (h(103) + 9) \bmod 10$$

 $h = 2$ , diletakkan pada indeks 2

→ insert 2

$$h = 2 \bmod 10$$

 $h = 2$ , karena indeks 2 sudah terisi maka lakukan 1<sup>st</sup> probe

$$h = (h(2) + 1) \bmod 10$$

 $h = 3$ , karena indeks 3 telah terisi maka lakukan 2<sup>nd</sup> probe

$$h = (h(2) + 4) \bmod 10$$

$h = 6$  . diletakkan pada indeks 6

Nama : Ruth Aulya Silalahi

NIM : 11S20018

Kelas : 12IF1

## Praktikum Algoritma dan Struktur Data

Method pada Test HashSet dan Test HashMap

1. Add (menambahkan elemen pada hashset)

```
ArrayList<Integer> arr = new ArrayList<Integer>();  
arr.add(1);  
arr.add(2);  
arr.add(3);  
arr.add(4);  
arr.add(5);
```

2. Put (menambahkan data(indeks, nilai) pada hashmap)

```
Map<Integer, String> map = new HashMap<Integer, String>();  
map.put(1, "Telletubbies");  
map.put(2, "Tinky Winky");  
map.put(3, "Dipsy");  
map.put(5, "Lala");  
map.put(6, "Poo");
```

3. Size (mengetahui ukuran elemen)

```
System.out.print("dilakukan uji method size() hset1: ");  
System.out.println(hset1.size());
```

```
    dilakukan uji method size() hset1: 5
```

4. Iterator (menampilkan nilai)

```
System.out.print("dilakukan uji method iterator() hset1: ");  
Iterator<Integer> itr = hset1.iterator();  
while(itr.hasNext()){  
    System.out.println(" " + itr.next() + " ");  
}
```

```
dilakukan uji method iterator() hset1: 1
2
3
4
5
```

5. Contains (memeriksa sebuah nilai ada atau tidak pada hset)

```
System.out.print("dilakukan uji method contains(100) hset0: ");
System.out.println(hset0.contains(100));
System.out.println(" ");
```

```
dilakukan uji method contains(100) hset0: true
```

```
System.out.print("dilakukan uji method contains(2) hset1: ");
System.out.println(hset1.contains(2));
```

```
dilakukan uji method contains(2) hset1: true
```

```
System.out.print("dilakukan uji method contains(100) hset1: ");
System.out.println(hset1.contains(100));
```

```
dilakukan uji method contains(100) hset1: false
```

6. ContainsKey (memeriksa apakah terdapat indeks atau tidak pada hmap)

```
System.out.print("dilakukan uji method containsKey(4) hmap1: ");
System.out.println(hmap1.containsKey(4));
System.out.println(" ");
```

```
System.out.print("dilakukan uji method containsKey(1) hmap1: ");
System.out.println(hmap1.containsKey(1));
System.out.println(" ");
```

```
dilakukan uji method containsKey(4) hmap1: false
```

```
dilakukan uji method containsKey(1) hmap1: true
```

## 7. Remove (menghapus elemen yang diinginkan)

```
System.out.print("dilakukan uji method put(7, Bayi Matahari) hmap1: ");
System.out.println(hmap1.put(7, "Bayi Matahari"));

System.out.print("dilakukan uji method toString() hmap1: ");
System.out.println(hmap1.toString());

        dilakukan uji method toString() hmap1: {1=Telletubbies, 2=Tinky Winky, 3=Dipsy, 5=Lala, 6=Poo, 7=Bayi Matahari}

System.out.print("dilakukan uji method remove(7) hmap1: ");
System.out.println(hmap1.remove(7));
System.out.println(" ");

        dilakukan uji method remove(7) hmap1: Bayi Matahari
```

## 8. getMatch (menampilkan nilai yang sama dari sebuah elemen)

```
System.out.print("dilakukan uji method getMatch(1) hset1: ");
System.out.println(hset1.getMatch(1));

        dilakukan uji method getMatch(1) hset1: 1
```

## 9. get (mendapatkan nilai dari sebuah indeks/key)

```
System.out.print("dilakukan uji method get(5) hmap1: ");
System.out.println(hmap1.get(5));

        dilakukan uji method get(5) hmap1: Lala
```

## 10. Clear (mengosongkan seluruh set)

```
System.out.print("dilakukan uji method clear() hmap0: ");
hmap0.clear();
```

## 11. isEmpty()

```
System.out.print("dilakukan uji method isEmpty() hmap0: ");
System.out.println(hmap0.isEmpty());

        dilakukan uji method isEmpty() hmap0: true
```

## 12. toString()

```
System.out.print("dilakukan uji method toString() hmap1: ");
System.out.println(hmap1.toString());

        dilakukan uji method toString() hmap1: {1=Telletubbies, 2=Tinky Winky, 3=Dipsy, 5=Lala, 6=Poo}
```