

# Modul 7 – Hashing

---

## Hashing

Buatlah fungsi-fungsi yang diperlukan dalam pencarian berdasarkan menggunakan konsep *hashing*, dan penanganan *collusion* dengan *chaining*. Fungsi-fungsi tersebut antara lain :

1. *Hash Function* yaitu fungsi hash yang menggunakan *remainder function*, dengan dua argument atau parameter, yaitu :
  - a. data yang akan dimasukkan ke dalam table hash
  - b. bilangan modulo

Sedangkan return value nya berupa adalah slot atau posisi data pada *hash table*

Contoh berikut adalah eksekusi reminderfunction dengan argument data berupa 55, dan bilangan modulo adalah 10, sehingga output yang dihasilkan adalah  $55 \% 10 = 5$

```
slot=remainderFunction(55,10)
print(slot)
```

5

2. *create Hash Table*, yaitu fungsi pembuatan *hash table*. Gunakan list 2D pada pembuatan hash table ini, karena dibutuhkan untuk penanganan *collusion* dengan *chaining*. Argumen yang terdapat pada fungsi ini adalah ukuran dari hash table, sedangkan return value berupa Hash table yang telah terbentuk (berupa list 2D, dengan nilai *default* adalah [None])

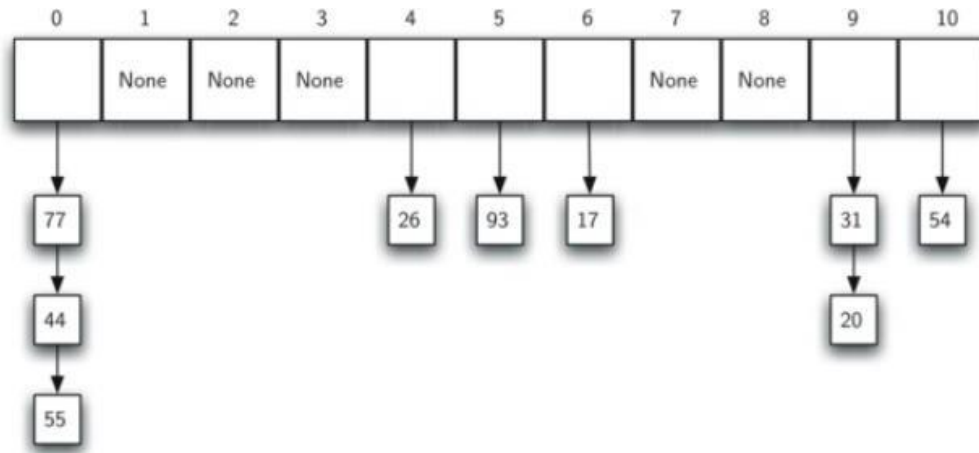
Berikut adalah inisialisasi hash table dengan ukuran 11 :

```
hashTable=createHashTable(11)
print(hashTable)
```

```
[[None], [None], [None], [None], [None], [None], [None], [None], [None], [None], [None]]
```

3. *chaining*, yaitu fungsi penempatan sejumlah data pada *hash table* yang telah dibuat. Fungsi chaining ini memiliki argument berupa list data (kumpulan data) yang akan ditempatkan pada hash table, dan hash table yang telah dibuat sebelumnya. Penempatan

data yang dibuat berdasarkan metode *chaining* untuk penanganan *collusion*. Jika suatu data memiliki nilai hash yang sama (berdasarkan fungsi hash, gunakan remainder function yang telah dibuat), maka data ditempatkan pada slot yang sama, akan tetapi pada indeks yang berbeda, seperti yang terlihat pada Gambar berikut :



Contoh eksekusi fungsi chaining ini dapat dilihat sebagai berikut (size hash table adalah 11, dan sudah dibuat sebelumnya dengan menggunakan create hash table):

```

: a=[54, 26, 93, 17, 77, 31,44,55,20]
: chaining(a,hashTable)
: print(hashTable)

[[77, 44, 55], [None], [None], [None], [26], [93], [17], [None], [None], [31, 20], [54]]

```

4. searchHash, yaitu fungsi pencarian berdasarkan fungsi hash yang telah dibuat sebelumnya. Argumen adalah data yang akan dicari dan hash table yang telah dibuat sebelumnya. Return value adalah False (jika data tidak ada), atau Nomor slot dan indeks jika data ada, seperti contoh berikut

```

: searchHash(66,hashTable)
: False

: searchHash(54,hashTable)
: ('data berada di slot ke-', 10, ' dan indeks ke-', 0)

: searchHash(20,hashTable)
: ('data berada di slot ke-', 9, ' dan indeks ke-', 1)

: searchHash(55,hashTable)
: ('data berada di slot ke-', 0, ' dan indeks ke-', 2)

: searchHash(100,hashTable)
: False

```