Modul 7 - Hashing

1. Hash Function menggunakan remainderFunction()

```
def remainderFunction(data, num):
    return data%num

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

```
5
```

2. Fungsi createHashTable

```
def createHashTable(num):
    temp = []
    for i in range(num):
        temp.append([None])
    return temp

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

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

3. Fungsi Chaining

```
def chaining(data, table):
    for i in range(len(data)):
        idx = remainderFunction(data[i], len(table))
        if table[idx][0] is None:
            table[idx][0] = data[i]
        else:
            table[idx].append(data[i])
    return table


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

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

4. Fungsi SearchHash

```
def searchHash(data, table):
    hashValue = remainderFunction(data, len(table))
    if data in table[hashValue]:
        slot = hashValue
        slotValue = table[hashValue]
        for index in range(len(slotValue)):
            if slotValue[index] == data:
                return f"data berada pada slot ke - {slot} dan indeks ke - {index}"
    else:
        return False


print(searchHash(66, hashTable))
```

```
False
```

```
print(searchHash(54, hashTable))
```

```
data berada pada slot ke - 10 dan indeks ke - 0
```

```
print(searchHash(20, hashTable))
```

```
data berada pada slot ke - 9 dan indeks ke - 1
```

```
print(searchHash(55, hashTable))
```

```
data berada pada slot ke - 0 dan indeks ke - 2
```

```
print(searchHash(100, hashTable))
```

```
False
```