

## 5\_2-BinarySearch

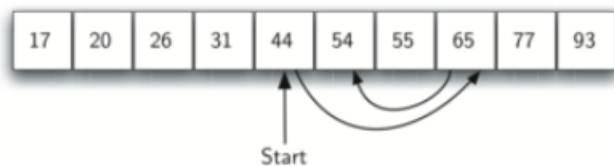
### 1 Pencarian/*Searching*

Binary Searching \*\*\*

Pencarian dengan algoritma *sequential search* akan menemukan data dengan cepat jika data yang dicari terletak di indeks pertama dari suatu list data. Akan tetapi jika data yang dicari berada di posisi terakhir dari suatu list atau posisi lain, maka algoritma *sequential search* ini akan membutuhkan waktu yang lebih lama, karena algoritma ini akan mencari satu persatu, dimulai dari indeks pertama sampai dengan indeks terakhir.

Pada algoritma **binary search** pencarian tidak dilakukan satu persatu, melainkan memanfaatkan kelebihan pencarian di data yang sudah terurut. Pada *binary search*, pencarian dilakukan pertama kali di indeks list yang berada ditengah. Jika data yang dicari sama dengan data pada indeks tersebut, maka pencarian berakhir. Akan tetapi jika data yang dicari lebih besar dari data yang berada di indeks tengah tersebut, maka dilakukan pencarian lagi. Hanya saja pencarian tidak dilakukan pada seluruh data, hanya data yang berada indeks-indeks setelah indeks tengah tadi, karena data sebelum indeks tengah, pasti lebih kecil nilainya, sehingga tidak perlu dilakukan pencarian lagi. Proses ini dilakukan secara terus menerus sampai data ketemu atau tidak ada lagi data yang dapat dicari di dalam list. Oleh karena itu, *binary search* tidak perlu mencari di semua data pada list, sehingga waktu komputasi yang dibutuhkan lebih cepat dibandingkan dengan pencarian *sequential search*.

Ilustrasi pencarian *binary search* ini dapat dilihat pada Gambar



berikut.

#### 1.1 Code

Berikut code untuk *binary search* dengan metode iteratif

```
In [1]: def binarySearch(listData, data):  
    first = 0  
    last = len(listData) - 1  
    found = False  
  
    while first <= last and not found:
```

```

midpoint = (first + last) // 2
if listData[midpoint] == data:
    found = True

else:
    if data < listData[midpoint]:
        last = midpoint - 1
    else:
        first = midpoint + 1

return found

```

```

In [3]: a=[4,6,10,34,56,78,99]
        print(binarySearch(a,34))

```

True

## 1.2 Latihan - 2

Modifikasi code Binary Search tersebut, sehingga return value dari fungsi tersebut adalah indeks data (jika data ada)

## 1.3 Code

Selain dengan model iteratif, algoritma binary search juga dapat ditulis secara rekursif, seperti yang ditunjukkan pada program berikut

```

In [4]: def binarySearch2(listData, data):
        if len(listData) == 0:
            return False
        else:
            midpoint = len(listData) // 2

            if listData[midpoint] == data:
                return True
            else:
                if data < listData[midpoint]:

                    return binarySearch2(listData[first:last+1], data)
                else:

                    return binarySearch2(listData[midpoint + 1:], data)

```

```

In [5]: a=[4,6,10,34,56,78,99]

```

```

In [6]: binarySearch2(a,78)

```

```

Out[6]: True

```

In [0]: