

MODUL VIII

SEARCHING

8.1 TUJUAN

Memahami dan menerapkan beberapa algoritma pencarian (searching) dalam menyelesaikan berbagai studi kasus.

8.2 DASAR TEORI

a. Konsep Dasar Searching

Searching adalah proses mendapatkan informasi berdasarkan kunci tertentu dari sejumlah informasi yang telah disimpan. Cara pencariannya adalah dengan cara mengecek datanya satu persatu secara berurutan. Jika data ditemukan, maka selesai.

b. Linear Search

Linear search merupakan program search yang mudah dipahami, linear search memiliki kelebihan apabila data yang di cari letaknya pada data – data awal sehingga prosesnya berjalan cepat, namun apabila data yang di cari letaknya pada data terakhir maka pencarian lebih memakan waktu yang cukup lama pula. karena di linear search mengunjungi setiap elemen data yang ada.

c. Binary Search

Binary search merupakan algoritma pencarian yang paling efisien. Algoritmanya adalah dengan membagi data menjadi dua bagian dengan syarat data tersebut telah terurut. Jika telah terurut maka perlu mencari indeks yang paling tengah dengan rumus:

$$\text{int middle} = \text{array.length}/2$$

Setelah menentukan indeks tengah maka ambil datanya. Jika kita akan mencari data yang lebih kurang dari data middle maka cari sebelah kiri, jika mencari data yang lebih besar dari data middle maka cari sebelah kanan.

8.3 LATIHAN

A. LINEAR SEARCH

```
def linear_search(keyword, data):
    for i in range(len(data)):
        if str(data[i]).lower() == keyword.lower():
            print(f"Keyword {keyword} has found at indeks {i}")
            return i
    print(f"Keyword {keyword} not found")
    return -1

data = [32, 7, 44, 21, 61, 25, 45]
keyword = input("Input keyword : ")
linear_search(keyword, data)
```

Output:

```
Input keyword : 21
Keyword 21 has found at indeks 3
```

 Jika angka yang dicari ada di data

```
Input keyword : 41
Keyword 41 not found
```

 Jika angka yang dicari tidak terdapat pada data

B. BINARY SEARCH

```
def bubble_sort(keyword, data):
    for i in range(len(data)):
        for j in range(len(data) - i - 1):
            if data[j] > data[j+1]:
                data[j], data[j+1] = data[j+1], data[j]
    return binary_search(keyword, data)

def binary_search(keyword, data):
    left = 0
    right = len(data) - 1

    while left <= right:
        mid = (left+right)//2
        if str(data[mid]).lower() > keyword.lower():
            right = mid - 1
        elif str(data[mid]).lower() < keyword.lower():
            left = mid + 1
        else:
            print(data)
            print(f"Keyword {keyword} has found at indeks {mid}")
            return mid

    print(f"Keyword {keyword} not found")
    return -1

data = [32, 7, 44, 21, 61, 25, 45]
keyword = input("Input keyword : ")
bubble_sort(keyword, data)
```

Output:

```
Input keyword : 32  
[7, 21, 25, 32, 44, 45, 61]  
Keyword 32 has found at indeks 3
```

Jika data yang dicari ada

```
Input keyword : 10  
Keyword 10 not found
```

Jika data yang dicari tidak ada

8.4 TUGAS

1. Pak polisi memiliki database yang berisi plat nomor mobil. Terdapat 10 nomor dalam database tersebut, yaitu sebagai berikut : *[R 2477 SR], [R 1234 DJ], [R 7015 LP], [R 0201 RR], [R 3304 DA], [R 2401 SK], [R 2103 RT], [R 1708 RI], [R 1111 SR], [R 4987 LH]*. Pada suatu hari pak polisi tersebut melihat kendaraan bernomor *[R 2488 SR]* berada di area larang parkir. Bantulah Pak Polisi tersebut untuk mengecek apakah nomor tersebut terdapat di dalam database atau tidak. Gunakan Algoritma Linear Search ! **(Skor 30)**
2. Dalam suatu kelas terdapat 14 mahasiswa yang memiliki nim sebagai berikut: *20103023, 20103002, 20103019, 20103001, 20103017, 20103005, 20103011, 20103003, 20103009, 20103021, 20103006, 20103015, 20103013, 20103007*. Dengan algoritma binary search, carilah nim 20103015 apakah berada di kelas tersebut atau tidak ! **(Skor 30)**
3. Pak Anto membuat program untuk meng-generate bilangan acak. Saat program dijalankan, program memberikan daftar bilangan acak sebagai berikut: *17, 2, 15, 7, 72, 31, 12, 57, 63, 71, 23, 92, 1*
Bantulah Pak Anto membuat program pencarian untuk bilangan acak tersebut dengan menggunakan Algoritma Binary Search. Angka yang dicari adalah 72! **(Skor 40)**

