

LAPORAN PRAKTIKUM
PEMROGRAMAN 1
MODUL VIII



Oleh:

Muhamad Luthfi Hamdani

2211104020

SE 06-A

PRODI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2023

I. 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.

II. GUIDED

A. Linear Search

Source code:

```
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 index {i}")  
            return i  
    print(f"Keyword {keyword} not found")  
    return -1  
  
data = [23, 3, 4, 78, 9, 10, 32]  
keyword = input("input keyword: ")  
linear_search( keyword, data)
```

Output :

Jika angka yang dicari ada pada data :

```
input keyword: 23  
Keyword '23' has found at index 0
```

Jika angka yang dicari tidak ada pada data:

```
input keyword: 233  
Keyword 233 not found
```

B. Binary Search

Source code:

```
def bubble_sort(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]

def binary_search(keyword, data):
    left = 0
    right = len(data) - 1
    while left <= right:
        mid = (left + right) // 2
        if data[mid] > keyword:
            right = mid - 1
        elif data[mid] < keyword:
            left = mid + 1
        else:
            print(data)
            print(f"Keyword {keyword} has been found at index {
mid}")
            return mid
    print(f"Keyword {keyword} not found")
    return -1

data = [23, 3, 4, 78, 9, 10, 32]
bubble_sort(data)

keyword = int(input("Input keyword: "))
binary_search(keyword, data)
```

Output:

Jika angka yang dicari ada pada data :

```
Input keyword: 9
[3, 4, 9, 10, 23, 32, 78]
Keyword 9 has been found at index 2
```

Jika angka yang dicari tidak ada pada data :

```
Input keyword: 100
Keyword 100 not found
```

III. UNGUIDED

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)**

Source code:

```
def linear_search(keyword, data):  
    for i in range(len(data)):  
        if data[i] == keyword:  
            return i  
    return -1  
  
database = ["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"]  
keyword = "R 2488 SR"  
  
result = linear_search(keyword, database)  
  
if result != -1:  
    print("Nomor plat ditemukan di indeks", result)  
else:  
    print("Nomor plat tidak ditemukan dalam database.")
```

Output :

```
Nomor plat tidak ditemukan dalam database.
```

kendaraan bernomor *[R 2488 SR]* yang berada di area larang parkir tidak ada dalam database

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)

Source code:

```
def binary_search(keyword, data):  
    left = 0  
    right = len(data) - 1  
  
    while left <= right:  
        mid = (left + right) // 2  
        if data[mid] == keyword:  
            return True  
        elif data[mid] < keyword:  
            left = mid + 1  
        else:  
            right = mid - 1  
  
    return False  
  
nim_list = [20103023, 20103002, 20103019, 20103001, 20103017,  
20103005, 20103011, 20103003, 20103009, 20103021, 20103006,  
20103015, 20103013, 20103007]  
keyword = 20103015  
  
nim_list.sort()  
  
if binary_search(keyword, nim_list):  
    print("NIM", keyword, "ditemukan di kelas.")  
else:  
    print("NIM", keyword, "tidak ditemukan di kelas.")
```

Output :

```
NIM 20103015 ditemukan di kelas.
```

Nim 20103015 ditemukan pada index 11

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)

Source code:

```
def binary_search(keyword, data):
    left = 0
    right = len(data) - 1
    while left <= right:
        mid = (left + right) // 2
        if data[mid] == keyword:
            return mid
        elif data[mid] < keyword:
            left = mid + 1
        else:
            right = mid - 1
    return -1

data = [17, 2, 15, 7, 72, 31, 12, 57, 63, 71, 23, 92, 1]
keyword = 72

data.sort()

result = binary_search(keyword, data)

if result != -1:
    print("Angka 72 ditemukan di indeks", result)
else:
    print("Angka 72 tidak ditemukan dalam daftar bilangan acak.")
```

Output :

```
Angka 72 ditemukan di indeks 11
```

Setelah diurutkan angka 72 dapat ditemukan pada indeks 11