

**LAPORAN PRAKTIKUM
PEMROGRAMAN 1
MODUL VIII**



Oleh:

NAMA: AHMAD JUNAIDI

NIM: 2211104002

KELAS: SE06A

**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

```
#Binary Search
def bubble_sort(data, keyword):
    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):
    sorted_data = data
    print(f'Data (sorted) = {sorted_data}')
    left = 0
    right = len(sorted_data) - 1
    while left <= right:
        mid = (left + right) // 2
        str_data = str(sorted_data[mid]).lower()
        if str_data > keyword.lower():
            right = mid - 1
        elif str_data < keyword.lower():
            left = mid + 1
        else:
            print(f'Keyword {keyword} has been found at index {mid}')
            return mid
    print(f'Keyword {keyword} has not been found')
    return -1

data = [23, 45, 4, 78, 10, 9]
keyword = input("Input keyword: ")
bubble_sort(data, keyword)
```

Output:

Jika angka yang dicari ada di data:

```
PS C:\Users\junai> & C:/Users/junai/AppData/
Input keyword: 23
Data (sorted) = [4, 9, 10, 23, 45, 78]
Keyword 23 has been found at index 3
PS C:\Users\junai> █
```

Jika angka yang dicari tidak terdapat pada data:

```
PS C:\Users\junai> & C:/Users/junai/AppData/
Input keyword: 44
Data (sorted) = [4, 9, 10, 23, 45, 78]
Keyword 44 has not been found
PS C:\Users\junai> █
```

B. BINARY SEARCH

```
#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 been found at index {i}")
            return i
    print(f"Keyword '{keyword}' has not been found")
    return -1

data = [23, 3, 4, 10, 32]
keyword = input("Input Keyword: ")
linear_search(keyword, data)
```

Output:

Jika data yang dicari ada:

```
PS C:\Users\junai> & C:/Users/junai/AppData/Local/Programs/Python/Python39-64/Scripts/python.exe C:/Users/junai/AppData/Local/Programs/Python/Python39-64/Scripts/python.exe
Input Keyword: 32
Keyword '32' has been found at index 4
PS C:\Users\junai>
```

Jika data yang dicari tidak ada:

```
PS C:\Users\junai> & C:/Users/junai/AppData/Local/Programs/Python/Python39-64/Scripts/python.exe C:/Users/junai/AppData/Local/Programs/Python/Python39-64/Scripts/python.exe
Input Keyword: 35
Keyword '35' has not been found
PS C:\Users\junai>
```

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)

Jawab:

Program:

```
def linear_search(search_input, database):
    for i in range(len(database)):
        if str(database[i][0]).lower() == search_input.lower():
            print(f"Plat dengan nomor '{search_input}'  
' ada pada database dengan index {i}")
            return i
    print(f"Plat dengan nomor '{search_input}' tidak ada pada database")
    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"]
]

search_input = input("Masukkan plat yang ingin dicari : ")
linear_search(search_input, database)
```

Output:

```
PS C:\Users\junai> & C:/Users/junai/AppData/Local/Programs/Python/Python38-32/python.exe C:/Users/junai/AppData/Local/Programs/Python/Python38-32/Scripts/Aufa Muhammad I_SE06A/task1.py
Masukkan plat yang ingin dicari : R 1234 DJ
Plat dengan nomer 'R 1234 DJ' ada pada database dengan index 1
PS C:\Users\junai>
```

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)

Jawab:

Program:

```
def bubble_sort(data, keyword):
    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):
    sorted_data = data
    print(f'Data (sorted) = {sorted_data}')
    left = 0
    right = len(sorted_data) - 1
    while left <= right:
        mid = (left + right) // 2
        str_data = str(sorted_data[mid]).lower()
        if str_data > keyword.lower():
            right = mid - 1
        elif str_data < keyword.lower():
            left = mid + 1
        else:
            print(f'Mahasiswa dengan NIM {keyword} terdapat di kelas")
            return mid
    print(f'Mahasiswa dengan NIM {keyword} tidak terdapat di kelas")
    return -1

data = [20103023, 20103002, 20103019, 20103001,
        20103017, 20103005, 20103011, 20103003,
        20103009, 20103021, 20103006, 20103015,
        20103013, 20103007]
keyword = input("Masukkan NIM yang ingin dicari : ")
bubble_sort(data, keyword)
```

Output:

```
Masukkan NIM yang ingin dicari : 20103015
Data (sorted) = [20103001, 20103002, 20103003, 20103005, 20103006,
03007, 20103009, 20103011, 20103013, 20103015, 20103017, 20103019,
03021, 20103023]
Mahasiswa dengan NIM 20103015 terdapat di kelas
PS C:\Users\junai>
```

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)

Jawab:

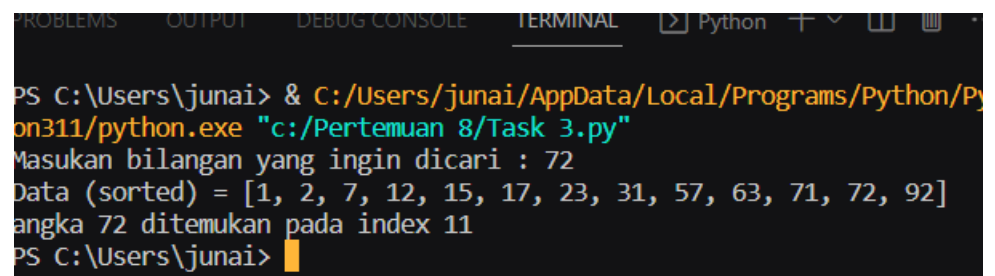
Program:

```
def bubble_sort(data, keyword):
    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):
    sorted_data = data
    print(f'Data (sorted) = {sorted_data}')
    left = 0
    right = len(sorted_data) - 1
    while left <= right:
        mid = (left + right) // 2
        str_data = str(sorted_data[mid]).lower()
        if str_data > keyword.lower():
            right = mid - 1
        elif str_data < keyword.lower():
            left = mid + 1
        else:
            print(f"angka {keyword} ditemukan pada index {mid}")
            return mid
    print(f"angka {keyword} tidak ditemukan")
    return -1

data = [17, 2, 15, 7, 72, 31, 12, 57, 63, 71, 23, 92, 1]
keyword = input("Masukan bilangan yang ingin dicari : ")
bubble_sort(data, keyword)
```

Output:



The screenshot shows a terminal window with the following content:

```
PS C:\Users\junai> & C:/Users/junai/AppData/Local/Programs/Python/Python311/python.exe "c:/Pertemuan 8/Task 3.py"
Masukan bilangan yang ingin dicari : 72
Data (sorted) = [1, 2, 7, 12, 15, 17, 23, 31, 57, 63, 71, 72, 92]
angka 72 ditemukan pada index 11
PS C:\Users\junai>
```