

**LAPORAN PRAKTIKUM
PEMROGRAMAN 1
MODUL 8**



Oleh:

Nama : Alfian Mutakim

Nim : 2211104024

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 mengunjungisetiap 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:

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

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

```
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 {i}")  
    return -1  
  
data = [23, 4, 5, 90, 18, 45]  
keyword = input("input keyword: ")  
linear_search(keyword, data)
```

Output yang dihasilkan :

Jika angka yang dicarai ada pada data :

```
on3.11.exe "C:/Users/User/Documents/  
input keyword: 90  
keyword 90 has found at index 3  
on3.11.exe "C:/Users/User/Documents/
```

Jika angka yang dicari tidak ada di data:

```
on3.11.exe "C:/Users/User/Documents/  
input keyword: 1  
keyword 1 not found  
PS C:\Users\User\Documents\Praktik
```

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 been found
at index {mid}")
            return mid

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

data = [23, 3, 4, 78, 9, 10, 32]
keyword = input("Input keyword: ")
bubble_sort(keyword, data)

```

Output:

Jika angka ada:

```

Input keyword: 10
[3, 4, 9, 10, 23, 32, 78]
Keyword '10' has been found at index 3
PS C:\Users\User\Documents\Praktikum Pemograman1\LapModu

```

Jika angka tidak ada:

```

Iain Natakini/binarysearch.py
Input keyword: 50
Keyword '50' not found
PS C:\Users\User\Documents\Praktikum Pemogr

```

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 !

Kode program:

```
def linear_search (keyword, data):  
    for i in range(len(data)):  
        if str(data[i]).lower() == keyword.lower():  
            print(f"No Kendaraan {keyword} ada pada data  
ke- {i}")  
            return i  
    print(f"No kendaraan {keyword} tidak ditemukan! ")  
    return -1  
  
data = ["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 = input("Masukkan No Kendaraan: ")  
linear_search(keyword, data)
```

Output yang dihasilkan:

```
013.11.exe C:/Users/User/Documents/Praktikum Pemrograman1/L  
Masukkan No Kendaraan: R 2488 SR  
No kendaraan R 2488 SR tidak ditemukan!  
PS C:\Users\User\Documents\Praktikum Pemrograman1\LapModul8
```

Kode program di atas menggunakan algoritma pencarian linier (linear search) untuk mencari keyword dalam array data. Algoritma ini bekerja dengan melakukan iterasi melalui setiap elemen array dan memeriksa apakah elemen tersebut sama dengan keyword yang dicari. Jika ditemukan, program mencetak pesan yang menyatakan di mana keyword ditemukan dan mengembalikan

indeksnya. Jika tidak ditemukan, program mencetak pesan bahwa keyword tidak ditemukan dan mengembalikan -1 sebagai hasil pencarian.

Dalam kasus diatas ditemukan bahwa No kendaraan yang dilihat polisi tidak ada dalam data base.

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 !

Kode program:

```
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"Mahasiswa dengan NIM '{keyword}'
Ditemukan di kelas!, pada indeks ke- {mid}")
            return mid

    print(f"Mahasiswa dengan NIM '{keyword}' Tidak
ditemukan dikelas")
```

```

        return -1

data = [220103023, 20103002, 20103019, 20103001,
20103017, 20103005, 20103011, 20103003, 20103009,
20103021, 20103006, 20103015, 20103013, 20103007]
keyword = input("Masukkan Nim: ")
bubble_sort(keyword, data)

```

Output yang dihasilkan:

```

an Mutakim/task2.py
Masukkan Nim: 20103015
[20103001, 20103002, 20103003, 20103005, 20103006
, 20103007, 20103009, 20103011, 20103013, 2010301
5, 20103017, 20103019, 20103021, 220103023]
Mahasiswa dengan NIM '20103015' Ditemukan di kela
s!, pada indeks ke- 9
PS C:\Users\User\Documents\Praktikum Pemograman1\

```

Program di atas merupakan implementasi algoritma Bubble Sort dan Binary Search dalam bahasa pemrograman Python. fungsi bubble_sort digunakan untuk mengurutkan data NIM dalam urutan numerik secara menaik.

Algoritma Bubble Sort bekerja dengan membandingkan pasangan elemen secara berurutan dan menukar posisinya jika urutannya salah. Proses pengurutan ini dilakukan untuk memastikan data terurut sehingga pencarian dapat dilakukan dengan menggunakan Binary Search.

Dalam implementasi data diatas, data NIM yang dicari adalah 20103015. Pengguna diminta untuk memasukkan NIM yang ingin dicari melalui input yang ditampung dalam variabel keyword. Kemudian, fungsi bubble_sort dan binary_search dipanggil dengan meneruskan keyword dan data sebagai argumen. Dari hasil pencarian nim tersebut ada di Keas!.

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!

Kode Program:

```
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"Angka '{keyword}' Ditemukan pada
indeks ke- {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("Masukkan Angka yang akan dicari: ")
bubble_sort(keyword, data)
```

Output yang dihasilkan:

```
PS C:\Users\User\Documents\Praktikum Pemograman1\LapModul1>
Masukkan Angka yang akan dicari: 72
[1, 2, 7, 12, 15, 17, 23, 31, 57, 63, 71, 72, 92]
Angka '72' Ditemukan pada indeks ke- 11
PS C:\Users\User\Documents\Praktikum Pemograman1\LapModul1>
```

Program di atas adalah implementasi dari algoritma Bubble Sort dan Binary Search dalam bahasa pemrograman Python. Tujuan

dari program ini adalah untuk mencari sebuah angka dalam sebuah list menggunakan metode pencarian biner setelah melakukan pengurutan angka-angka tersebut dengan metode Bubble Sort. Dari data yang akan dicari pak Anto diatas ditemukan bahwa no 72 ada pada indeks ke 11.