

LAPORAN PRAKTIKUM PEMROGRAMAN 1



Disusun Oleh:

Nama : Septiandi Nugraha

NIM : 21104060

Kelas : SE05 B

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

TAHAP 8

SEARCHING

7.1 TUJUAN

1. Mampu memahami dan menerapkan beberapa algoritma pencarian (searching) dalam menyelesaikan berbagai studi kasus

7.2 DASAR TEORI

1. Konsep Dasar Searching

Searching atau dalam bahasa Indonesia adalah pencarian merupakan sebuah proses untuk mendapatkan informasi berdasarkan kunci tertentu dari sejumlah informasi yang telah disimpan. Cara pencariannya yaitu dengan cara memeriksa data satu persatu secara berurutan. Jika data ditemukan, maka proses akan selesai.

2. Linear Search

Linear search merupakan program search yang mudah untuk dipahami, dalam linear search memiliki kelebihan apabila suatu data yang di cari letaknya pada data-data awal sehingga prosesnya berjalan cepat. Namun apabila data yang dicari letaknya pada data terakhir, maka pencarian lebih memakan waktu yang cukup lama pula. Karena di linear search mengunjungi setiap elemen data yang ada.

3. Binary Search

Binary search ini merupakan algoritma pencarian yang paling efisien di antara algoritma lain. Yaitu 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 data sebelah kiri. Jika mencari data yang lebih besar dari data middle maka cari sebelah kanan.

7.3 LATIHAN

1. Linear Search

```
def linear_search(data, keyword):  
    for i in range(len(data)):  
        if str(data[i]).lower() == keyword.lower():  
            print(f"Keyword {keyword} is found at index {i}")  
            return i  
  
    print("Error 404: Keyword {keyword} not found!")  
    return -1  
  
def insertion_sort(data):  
    for i in range(1, len(data)):  
        item = data[i]  
        j = i - 1  
  
        while j >= 0 and data[j] > item:  
            data[j + 1] = data[j]  
            j -= 1  
  
        data[j + 1] = item  
  
    return data  
  
def convert_data_ke_string(data):  
    convert_data = []  
  
    for i in data:  
        convert_data.append(str(i).lower())  
  
    return convert_data
```

Data yang tersedia:

```
data = [409, 509, 14, "Siti", "Maimunah", "Haji Maun", 15, 67, 89, 74]
keyword = input("Search something: ")
linear_search(data, keyword)
```

Ketika program di run untuk mencari data “Siti” dan “509” maka akan seperti ini:

```
Search something: siti
Keyword siti is found at index 3
```

```
Search something: 509
Keyword 509 is found at index 1
```

2. Binary Search

```
def insertion_sort(data):  
    for i in range(1, len(data)):  
        item = data[i]  
        j = i - 1  
        while j >= 0 and data[j] > item:  
            data[j + 1] = data[j]  
            j -= 1  
        data[j + 1] = item  
    return data  
  
def convert_data_ke_string(data):  
    convert_data = []  
    for i in data:  
        convert_data.append(str(i).lower())  
    return convert_data  
  
def binary_search(keyword, data):  
    converted_data = convert_data_ke_string(data)  
    sorted_data = insertion_sort(converted_data)  
    low = 0  
    high = len(data) - 1  
    while low <= high:  
        mid = (low + high) // 2  
        if sorted_data[mid] == keyword.lower():  
            print(f"Keyword '{keyword}' is found at index {mid}")  
            return mid  
        elif str(sorted_data[mid]) < str(keyword).lower():  
            low = mid + 1  
        else:  
            high = mid - 1  
    print(f"Error: Keyword '{keyword}' is not found!")  
    return -1
```

Data yang tersedia:

```
data = [409, 509, 14, "Siti", "Maimunah", "Haji Maun", 15, 67, 89, 74]
keyword = input("Search something: ")
binary_search(keyword, data)
```

Ketika program di run untuk mencari data “Siti” dan “509” maka akan seperti ini:

```
Search something: siti
Keyword 'siti' is found at index 9
```

```
Search something: 509
Keyword '509' is found at index 3
```

7.4 TUGAS

- a. Pak polisi memiliki database yang berisi plat nomor mobil. Terdapat 10 nomor dalam database tersebut, yaitu sebagai berikut : [R 300 SR], [R 1234 DJ], [R 701 LP], [R 3218 RR], [R 007 TU], [R 3 ST], [R 999 RT], [R 210 RO], [R 1111 II], [R 4987 LH] Pada suatu hari pak polisi tersebut melihat kendaraan bernomor [R 999 RS] berada di area larang parkir. Bantulah Pak Polisi tersebut untuk mengecek apakah nomor tersebut terdapat di dalam database atau tidak. Gunakan Algoritma Sequential Search !

Code:

```
import os
os.system('cls')

def linear_search(keyword, data):
    print(f>Data: {data}")
    for i in range(len(data)):
        if str(data[i]).lower() == keyword.lower():
            print(f"Nomor plat kendaraan '{keyword}' ditemukan dalam database index {i}")
            return i
    print(f"Nomor plat kendaraan '{keyword}' tidak ditemukan dalam database!")
    return -1

data = ['R 300 SR', 'R 1234 DJ', 'R 701 LP', 'R 3218 RR', 'R 007 TU', 'R 3 ST', 'R 999 RT', 'R 210 RO', 'R 1111 II', 'R 4987 LH']
keyword = input("Search something: ")
linear_search(keyword, data)
```

Jalankan:

```
Search something: R 999 RS
Data: ['R 300 SR', 'R 1234 DJ', 'R 701 LP', 'R 3218 RR'], 'R 007 TU', 'R 3 S
T', 'R 999 RT', 'R 210 R0', 'R 1111 II', 'R 4987 LH']
Nomor plat kendaraan 'R 999 RS' tidak ditemukan dalam database!
PS C:\Kuliah\Semester 2\Praktikum Pemrograman 1\Pertemuan 8 Sesudah UTS\Code
> █
```

- b. Dalam suatu kelas terdapat 13 mahasiswa yang memiliki nim sebagai berikut :
12102002, 121002004, 12102001, 12102003, 12102005, 12102008,
12102007, 12102006, 12102009, 121020013, 12102010, 12102012,
12102011. Dengan algoritma binary search, carilah nim 12102001 apakah
berada di kelas tersebut atau tidak !

Code:

```
import os
os.system('cls')

def insertion_sort(data):
    for i in range(1, len(data)):
        item = data[i]
        j = i - 1

        while j >= 0 and data[j] > item:
            data[j + 1] = data[j]
            j -= 1

        data[j + 1] = item

    return data

def convert_data_to_string(data):
    converted_data = []

    for i in data:
        converted_data.append(str(i).lower())

    return converted_data

def binary_search(keyword, data):
    converted_data = convert_data_to_string(data)
    sorted_data = insertion_sort(converted_data)
    print(f"Data (sorted): {sorted_data}")
    left = 0
    right = len(data) - 1

    while left <= right:
        mid = (left + right) // 2

        if sorted_data[mid] == keyword.lower():
            print(f"NIM '{keyword}' ditemukan dalam kelas. Pada index ke-{mid}")
            return mid
        elif str(sorted_data[mid]) < str(keyword).lower():
            left = mid + 1
        else:
            right = mid - 1

    print(f"NIM '{keyword}' tidak ditemukan dalam kelas!")
    return -1

data = [12102002, 121002004, 12102001, 12102003, 12102005, 12102008, 12102007,
12102006, 12102009, 121020013, 12102010, 12102012, 12102011]
keyword = input("Search something: ")
binary_search(keyword, data)
```


Jalankan:

```
Search something: 12102001
Data (sorted): ['121002004', '12102001', '121020013', '12102002', '12102003',
, '12102005', '12102006', '12102007', '12102008', '12102009', '12102010', '1
2102011', '12102012']
NIM '12102001' ditemukan dalam kelas. Pada index ke-1
PS C:\Kuliah\Semester 2\Praktikum Pemrograman 1\Pertemuan 8 Sesudah UTS\Code
> █
```

- c. Pak Anto membuat program untuk meng-generate bilangan acak. Saat program dijalankan, program memberikan daftar bilangan acak sebagai berikut : 21, 61, 28, 72, 44, 68, 37, 52, 75, 71 Bantulah Pak Anto membuat program pencarian untuk bilangan acak tersebut dengan menggunakan Algoritma Binary Search. Angka yang dicari adalah 71 !

Code:

```
import os
os.system('cls')

def insertion_sort(data):
    for i in range(1, len(data)):
        item = data[i]
        j = i - 1

        while j >= 0 and data[j] > item:
            data[j + 1] = data[j]
            j -= 1

        data[j + 1] = item

    return data

def convert_data_to_string(data):
    converted_data = []

    for i in data:
        converted_data.append(str(i).lower())

    return converted_data

def binary_search(keyword, data):
    converted_data = convert_data_to_string(data)
    sorted_data = insertion_sort(converted_data)
    print(f"Data (sorted): {sorted_data}")
    left = 0
    right = len(data) - 1

    while left <= right:
        mid = (left + right) // 2

        if sorted_data[mid] == keyword.lower():
            print(f"Data '{keyword}' ditemukan pada index ke-{mid}")
            return mid
        elif str(sorted_data[mid]) < str(keyword).lower():
            left = mid + 1
        else:
            right = mid - 1

    print(f"Data '{keyword}' tidak ditemukan!")
    return -1

data = [21, 61, 28, 72, 44, 68, 37, 52, 75, 71]
keyword = input("Search something: ")
binary_search(keyword, data)
```

Jalankan:

```
Search something: 71
Data (sorted): ['21', '28', '37', '44', '52', '61', '68', '71', '72', '75']
Data '71' ditemukan pada index ke-7
PS C:\Kuliah\Semester 2\Praktikum Pemrograman 1\Pertemuan 8 Sesudah UTS\Code
> █
```