

MODUL 4

Searching (Pencarian)

1. Tujuan

- Mahasiswa dapat menjelaskan prinsip beberapa algoritma searching.
- Mahasiswa dapat mengimplementasikan algoritma searching pada pembuatan sebuah aplikasi

2. Langkah-langkah Praktikum

Buatlah sebuah project baru pada java dengan nama Searching.

a. Algoritma Linear Search

Buat sebuah fungsi baru pada class Search sebagai berikut:

```
public static void linearSearch(int[] data, int cari) {
    boolean found = false;
    for (int i = 0; i < data.length; i++) {
        if (data[i] == cari) {
            found = true;
            break;
        }
    }
    if (found) {
        System.out.println("Data ditemukan");
    } else {
        System.out.println("Data tidak ditemukan");
    }
}
```

Penjelasan:

Kata kunci static pada pendeklarasian fungsi menandakan bahwa saat program di running, fungsi akan langsung dialokasikan di memory komputer. Hal ini juga mengindikasikan bahwa fungsi dapat digunakan dan dipanggil tanpa melalui instansiasi objek.

Tambahkan method main berikut untuk melakukan pengujian method:

```
public static void main(String[] args) {
    int[] data = {1, 5, 9, 10, 23, -1, 4};
    linearSearch(data, 5);
}
```

Task 1. Simpan dan jalankan program. Lampirkan screen capture coding dan hasil running program pada laporan! Beri penjelasan singkat menggunakan bahasa Anda sendiri mengenai cara kerja algoritma Linear Search!

Task 2. Modifikasilah method `linearSearch` agar dapat menampilkan juga pada index ke-berapa data ditemukan (apabila ditemukan)! Lampirkan script dan hasil pada laporan!

b. Algoritma Binary Search

Task 3. Tuliskan sebuah method pada class `Searching` untuk melakukan sorting data! Algoritma bebas.

Tambahkan algoritma berikut pada class `Searching`!

```
public static int binarySearch(int[] data, int left, int right, int cari) {
    if (right >= 1) {
        int middle = left + (right - left) / 2;
        if (data[middle] == cari) {
            return middle;
        } else if (data[middle] < cari) {
            return binarySearch(data, middle + 1, right, cari);
        } else {
            return binarySearch(data, left, middle - 1, cari);
        }
    }
    return -1;
}
```

Task 4. Buatlah script untuk pemanggilan method `binarySearch` pada method main dan lampirkan script serta hasilnya pada laporan!

Task 5. Lampirkan method `binarySearch` pada laporan, serta berilah penjelasan terkait alur kerja binary search!

c. Algoritma Interpolation Search

Tambahkan method berikut pada class `Searching`!

```
public static boolean interpolationSearch(int[] data, int left, int right, int cari) {
    int pos;
    boolean found = false;
    if (left <= right && cari >= data[left] && cari <= data[right]) {
        pos = left + (((right - left) / (data[right] - data[left])) * (cari - data[left]));
        if (data[pos] == cari) {
            found = true;
        } else if (data[pos] < cari) {
            return interpolationSearch(data, pos + 1, right, cari);
        } else {
            return interpolationSearch(data, left, pos - 1, cari);
        }
    }
    return found;
}
```

Isilah method main sebagai berikut!

```
public static void main(String[] args) {  
    int[] data = {1, 5, 9, 10, 23, -1, 4};  
    bubbleSort(data);  
    if(interpolationSearch(data, 0, data.length-1, 5)){  
        System.out.println("Data ditemukan");  
    }else{  
        System.out.println("Data tidak ditemukan");  
    }  
}
```

Task 6. Lampirkan script method interpolationSearch dan main dalam laporan, serta berilah penjelasan singkat terkait dengan algoritma interpolation search sesuai bahasa Anda sendiri!

3. Tugas Praktikum

- Buatlah sebuah array 1D bernama yangDiCari yang digunakan untuk menyimpan beberapa data yang akan dicari (minimum 3 angka)
- Buatlah sebuah array 1D lainnya yang berisikan 10 buah data angka
- Buatlah sebuah array 2D yang berisi kumpulan data angka (ukuran array sesuai kreatifitas masing-masing)
- Buatlah method interpolationSearch yang digunakan untuk melakukan pencarian setiap data yang ada dalam array yangDiCari pada kumpulan angka pada array 1D
- Buatlah method linearSearch yang digunakan untuk melakukan pencarian setiap data yang ada dalam array yangDiCari pada array 2D

Contoh:

yangDiCari = {3, 6, 11}

data1D = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

data2D = {{1, 4, 6}, {2, 1, 7}, {3, 4, 11}};

Pencarian...

Angka 3

Pada array 1D = ditemukan

Pada array 2D = ditemukan

Angka 6

Pada array 1D = ditemukan

Pada array 2D = ditemukan

Angka 11

Pada array 1D = tidak ditemukan

Pada array 2D = ditemukan

