

LAPORAN PRAKTIKUM
Algoritma dan Struktur Data

PRAKTIKUM 6 W7
Searching



Oleh:
JESSICA AMELIA
2341760185
SIB 1A/16

PROGRAM STUDI SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2023/2024

Jobsheet 6

Percobaan 1 Sequential Search

Pertanyaan

1. Amati output dari program. Apakah algoritma searching dengan sequential search sudah diimplementasikan dengan tepat? Jika tidak, lakukan modifikasi untuk memperbaiki kode program.

Jawab : Belum tepat, karena kondisi dalam if mengecek i apakah sama dengan key, bukan apakah elemen array pada indeks i sama dengan key. Seharusnya Anda mengecek `arr[i] == key`.

```
package PRAKTIKUM06;

public class Sorting {
    public static void sequentitalSearch(int[] arr, int key){
        int hasil = -1;
        for (int i=0; i<arr.length; i++){
            if(arr[i]==key){
                hasil = i;
                break;
            }
        }
        if(hasil != -1){
            System.out.println("Data ditemukan pada indeks ke-" + hasil);
        }else {
            System.out.println(x:"Data tidak ditemukan");
        }
    }
}

Run | Debug
public static void main(String[] args) {
    int[] daftarNilai = {10,5,20,15,80,45};
    sequentitalSearch(daftarNilai, key:5);
}

D:\Roaming\Code\User\workspacestorage\
Data ditemukan pada indeks ke-1
PS D:\.1\Semester 2\ASD PRAKTIKUM\ASD>
```

2. Lakukan perubahan array `daftarNilai` pada fungsi `main()`. Jika terdapat lebih dari 1 data pada array, indeks mana yang akan menjadi output? Mengapa demikian?

```
Run | Debug
public static void main(String[] args) {
    int[] daftarNilai = { 10, 5, 20, 15, 5, 45 };
    sequentitalSearch(daftarNilai, 5);
}
```

Jawab : Indeks yang akan menjadi output adalah indeks dari kemunculan pertama dari nilai dalam array. Karena implementasi sequential search mencari elemen target dalam array dan mengembalikan indeks dari kemunculan pertama dari elemen tersebut.

3. Modifikasi kode program untuk menampilkan indeks di mana data pertama kali ditemukan

Jawab :

```
package PRAKTIKUM06;

public class Sorting {
    public static void sequentialSearch(int[] arr, int key){
        int hasil = -1;
        for (int i=0; i<arr.length; i++){
            if(arr[i]==key){
                hasil = i;
                break;
            }
        }
        if(hasil != -1){
            System.out.println("Data " + key + " ditemukan pada indeks ke-" + hasil);
        }else {
            System.out.println("Data " + key + " tidak ditemukan");
        }
    }
}

Run | Debug
public static void main(String[] args) {
    int[] daftarNilai = {10,5,20,15,5,45};
    sequentialSearch(daftarNilai, key:5);
}
}
```

```
pData\Roaming\Code\User\workspaceStorage\el
Data 5 ditemukan pada indeks ke-1
PS D:\.1\Semester 2\ASD PRAKTIKUM\ASD>
```

4. Jika data terurut secara ascending, modifikasi apa yang perlu dilakukan terhadap method sequentialSearch()?
Jawab : Tidak memerlukan modifikasi, karena sequentialSearch tidak memperdulikan urutan dalam mencari letak data pada indeks.
5. Jika data terurut secara descending, modifikasi apa yang perlu dilakukan terhadap method sequentialSearch()?
Jawab : Tidak memerlukan modifikasi, karena sequentialSearch tidak memperdulikan urutan dalam mencari letak data pada indeks.
6. Modifikasi method sequentialSearch() sehingga memiliki return type int dan mengembalikan nilai indeks di mana data ditemukan
Jawab :

```

TIKUM06 > Sorting.java > Sorting > main(String[])
package PRAKTIKUM06;

public class Sorting {
    public static int sequentialSearch(int[] arr, int key){
        int hasil = -1;
        for (int i=0; i<arr.length; i++){
            if(arr[i]==key){
                hasil = i;
                break;
            }
        }
        return hasil;
    }
    Run | Debug
    public static void main(String[] args) {
        int[] daftarNilai = {10,5,20,15,5,45};
        int key =5;
        int hasilIndeks = sequentialSearch(daftarNilai, key);
        if(hasilIndeks != -1){
            System.out.println("Data " + key + " ditemukan pada indeks ke-" + hasilIndeks);
        }else {
            System.out.println("Data " + key + " tidak ditemukan");
        }
    }
}

```

Percobaan 2 Binary Search

Pertanyaan

1. Jika data tidak terurut, apakah method `binarySearchAsc()` dapat digunakan?
Jawab : Tidak dapat digunakan, karena binary search bergantung pada sifat data yang terurut untuk menargetkan elemen tengah dan membagi ruang pencarian menjadi dua bagian yang sama besar.
2. Tambahkan method `binarySearchDesc()` untuk melakukan pencarian data dengan binary search terhadap array of integer yang terurut secara descending

Jawab :

```

public static int binarySearchDesc(int[] arr, int key){
    int start = 0, end = arr.length -1;

    while (start <= end){
        int mid = start + (end - start)/2;

        if (arr[mid] == key){
            return mid;
        }
        if (arr[mid]< key){
            end = mid -1;
        }
        else{
            start = mid + 1;
        }
    }
    return -1;
}

```

```

int[] descendingSortedNilai = {35, 30, 25, 20, 15, 10, 5};
int indexDesc = binarySearchDesc(descendingSortedNilai, key:20);
if(indexDesc != -1){
    System.out.println("Data ditemukan pada indeks ke-" + indexDesc + " dalam array yang terurut secara descending.");
}else {
    System.out.println("Data tidak ditemukan dalam array yang terurut secara descending.");
}
}

```

```
Data ditemukan pada indeks ke-3 dalam array yang terurut secara descending.  
PS D:.\1\Semester 2\ASD PRAKTIKUM\ASD>
```

3. Misalnya data input dapat dipastikan sudah terurut, tetapi tidak diketahui apakah data terurut secara ascending atau descending. Buat method `binarySearch()` dan tambahkan logika untuk memilih method yang tepat.

Jawab :

```
public static int binarySearch (int [] arr, int key){  
    if (arr.length == 0) {  
        return -1;  
    }  
  
    if (arr[0] < arr[arr.length - 1]) {  
        // Data terurut secara ascending  
        return binarySearchAsc(arr, key);  
    } else {  
        // Data terurut secara descending  
        return binarySearchDesc(arr, key);  
    }  
}
```

```
// binary search  
int[] sortedValues = {45, 35, 30, 25, 20, 15, 10, 5};  
int indexBinary = binarySearch(sortedValues, key:20);  
if(indexBinary != -1){  
    System.out.println("Data ditemukan pada indeks ke-" + indexBinary);  
}else {  
    System.out.println(x:"Data tidak ditemukan");  
}  
}
```

```
Data ditemukan pada indeks ke-4  
PS D:.\1\Semester 2\ASD PRAKTIKUM\ASD>
```

Tugas

Modifikasi tugas pada praktikum sorting yaitu class `Kelas.java`.

1. Buat method `sequentialSearchByNama()` untuk melakukan pencarian berdasarkan nama mahasiswa dengan algoritma sequential search. Gunakan method `contains()` agar keyword tidak harus sama persis dengan nama mahasiswa

```

public void sequentialSearchByNama(String keyword) {
    boolean ditemukan = false;
    for (Mahasiswa mahasiswa : daftarMahasiswa) {
        if (mahasiswa != null && mahasiswa.nama.contains(keyword)) {
            System.out.println("x: Mahasiswa yang dicari:");
            System.out.println("Nama: " + mahasiswa.nama);
            System.out.println("Umur: " + mahasiswa.umur);
            System.out.println("IPK: " + mahasiswa.ipk);
            ditemukan = true;
        }
    }
    if (!ditemukan) {
        System.out.println("Tidak ada mahasiswa dengan nama\"" + keyword + "\".");
    }
}

```

2. Buat method `binarySearchByUmur()` untuk melakukan pencarian berdasarkan umur mahasiswa dengan algoritma binary search

```

public int binarySearchByUmur(int umur) {
    int low = 0;
    int high = daftarMahasiswa.length - 1;

    while (low <= high) {
        int mid = low + (high - low) / 2;
        int midUmur = daftarMahasiswa[mid].umur;

        if (umur == midUmur) {
            System.out.println("Mahasiswa dengan umur " + umur + ":");
            System.out.println("Name: " + daftarMahasiswa[mid].nama);
            System.out.println("IPK: " + daftarMahasiswa[mid].ipk);
            return mid;
        } else if (umur < midUmur) {
            high = mid - 1;
        } else {
            low = mid + 1;
        }
    }

    System.out.println("Mahasiswa dengan " + umur + " tidak ditemukan.");
    return -1;
}

```

3. Coba gunakan method `sequentialSearchByNama()` dan `binarySearchByUmur()` pada fungsi `main()`.

```

public class Demo {
    Run | Debug
    public static void main(String[] args) {
        Kelas kelas1A = new Kelas(kuota:5);
        Mahasiswa mhs1 = new Mahasiswa(nama:"Ani", umur:18, ipk:4.00);
        Mahasiswa mhs2 = new Mahasiswa(nama:"Budi", umur:19, ipk:3.50);
        Mahasiswa mhs3 = new Mahasiswa(nama:"Cica", umur:17, ipk:3.75);
        Mahasiswa mhs4 = new Mahasiswa(nama:"Deni", umur:20, ipk:3.15);
        Mahasiswa mhs5 = new Mahasiswa(nama:"Eka", umur:18, ipk:3.00);

        kelas1A.add(mhs1);
        kelas1A.add(mhs2);
        kelas1A.add(mhs3);
        kelas1A.add(mhs4);
        kelas1A.add(mhs5);

        kelas1A.bubbleSortByIPK();
        kelas1A.selectionSortByUmur();
        kelas1A.insertionSortByIPKDesc();
        kelas1A.displayInfo();
        kelas1A.sequentialSearchByNama(keyword:"Deni");

        int searchIndexByAge = kelas1A.binarySearchByUmur(umur:20);
        if (searchIndexByAge != -1) {
            System.out.println("Umur ditemukan pada index: " + searchIndexByAge);
        } else {
            System.out.println(x:"Umur tidak ditemukan");
        }
    }
}

```

```

D:\Data\Roaming\Code\User\workspacestorage\eb00
Ani (umur : 18 IPK : 4.0)
Cica (umur : 17 IPK : 3.75)
Budi (umur : 19 IPK : 3.5)
Deni (umur : 20 IPK : 3.15)
Eka (umur : 18 IPK : 3.0)
Mahasiswa yang dicari:
Nama: Deni
Umur: 20
IPK: 3.15
Mahasiswa dengan umur 20:
Name: Deni
IPK: 3.15
Umur ditemukan pada index: 3
PS D:\.1\Semester 2\ASD PRAKTIKUM\ASD>

```

Catatan:

- Output yang diharapkan adalah index, nama, umur, dan IPK. Jika perlu, tambahkan method `displayInfo(int index)` pada class `Kelas` untuk menampilkan informasi mahasiswa pada indeks tertentu
- Jika terdapat lebih dari 1 mahasiswa dengan umur sesuai keyword, tampilkan semua data