

Data Structure and Algorithm Practicum

Searching



Name

Muhammad Baihaqi Aulia Asy'ari

NIM

2241720145

Class

1I

Department

Information Technology

Study Program

D4 Informatics Engineering

1.2 Sequential Search Method

1.2.1 Steps

1. Create a new project in NetBeans called `TestSearching`
2. Then, create a new package `week7`.
3. Create new `Students` class, then declare following attributes:

```
public class Students {  
    int nim, age;  
    String name;  
    double gpa;  
}
```

4. Create a constructor in `Students` class with parameters (`int ni`, `String nm`, `int age`, `double gpa`). Convert it to program code as follows:

```
public Students(int nim, int age, String name, double gpa) {  
    this.nim = nim;  
    this.age = age;  
    this.name = name;  
    this.gpa = gpa;  
}
```

5. Create `display()` method with `void` as its return type

```
public void display() {  
    System.out.println("NIM =" + nim);  
    System.out.println("Name =" + name);  
    System.out.println("Age =" + age);  
    System.out.println("GPA =" + gpa);  
}
```

6. Create a new `SearchStudent` class as follows.

```
public class SearchStudent {  
    Students[] listStd = new Students[5];  
    int idx;  
}
```

7. Create method `add()` at that class! This will be used for adding objects from `Students` class to `listStd` attribute

```

public void add(Students std) {
    if (idx < listStd.length) {
        listStd[idx] = std;
        idx++;
    } else {
        System.out.println("Data is already full");
    }
}

```

8. Create method `display()` in class `SearchStudent`! This `display()` method will be used to print all students data available in this class. Pay attention on how we use `for` loops differently. Even so, the concepts is still the same

```

public void display() {
    for (Students students : listStd) {
        students.display();
        System.out.println("-----");
    }
}

```

9. Create method `FindSeqSearch` with integer as its return type. Then fill in the function with sequential search algorithm.

```

public int findSeqSearch(int search) {
    int potition = 1;
    for (int i = 0; i < listStd.length; i++) {
        if (listStd[i].nim == search) {
            potition = i;
            break;
        }
    }
    return potition;
}

```

10. Create method `displayPosition` with void as its return type. And write these following code as follows

```

public void showPotition(int x, int pos) {
    if (pos != 1) {
        System.out.println("Data : " + x + " is found in index-"
            + pos);
    } else {
        System.out.println("Data : " + x + "is not found");
    }
}

```

-
11. Create method `displayData` with `void` as its return type. And write these following code as follows

```
public void showData(int x, int pos) {
    if (pos != 1) {
        System.out.println("NIM \t : " + x);
        System.out.println("Name \t : " + listStd[pos].name);
        System.out.println("Age \t : " + listStd[pos].age);
        System.out.println("IPK \t : " + listStd[pos].gpa);
    } else {
        System.out.println("Data " + x + " is not found");
    }
}
```

12. Create a main class named `StudentsMain` and add main method as follows

```
public class StudentsMain {
    public static void main(String[] args) {

    }
}
```

13. In main method, instantiate an object in `SearchStudent` that consist of 5 Students, then add all students object by calling add function in object `SearchStudent`

```
Scanner s = new Scanner(System.in);
Scanner sl = new Scanner(System.in);

SearchStudent data = new SearchStudent();
int amountStudent = 5;

System.out.println("-----");
System.out.println("Input student data accordingly from samllest
↪ NIM");
for (int i = 0; i < amountStudent; i++) {
    System.out.println("-----");
    System.out.print("NIM\t: ");
    int nim = s.nextInt();
    System.out.print("Name\t: ");
    String name = sl.nextLine();
    System.out.print("Age\t: ");
    int age = s.nextInt();
    System.out.print("GPA\t: ");
    double gpa = s.nextDouble();
}
```

```

        Students std = new Students(nim, age, name, gpa);
        data.add(std);
    }

```

14. Add method display to print all inserted data

```

System.out.println("-----");
System.out.println("Entire Student Data");
data.display();

```

15. To search students by their NIM, create a search variable to hold input from user. Then call method FindSeqSearch with its parameter is the search variable we've declared before

```

System.out.println("-----");
System.out.println("-----");
System.out.print("Search student by NIM: ");
int search = s.nextInt();
System.out.println("Using Sequential Search");
int potition = data.findSeqSearch(search);

```

16. Call method displayPosition from class SearchStudent.

```

data.showPotition(search, potition);

```

17. Call method displayData from class SearchStudent

```

data.showData(search, potition);

```

18. Run the program and see the result

```

package week7;

public class Students {
    int nim, age;
    String name;
    double gpa;

    public Students(int nim, int age, String name, double gpa) {
        this.nim = nim;
        this.age = age;
        this.name = name;
        this.gpa = gpa;
    }
}

```

```

        public void display() {
            System.out.println("NIM =" + nim);
            System.out.println("Name =" + name);
            System.out.println("Age =" + age);
            System.out.println("GPA =" + gpa);
        }
    }

package week7;

public class SearchStudent {
    Students[] listStd = new Students[5];
    int idx;
    public void add(Students std) {
        if (idx < listStd.length) {
            listStd[idx] = std;
            idx++;
        } else {
            System.out.println("Data is already full");
        }
    }

    public void display() {
        for (Students students : listStd) {
            students.display();

            → System.out.println("-----");
        }
    }

    public int findSeqSearch(int search) {
        int potition = 1;
        for (int i = 0; i < listStd.length; i++) {
            if (listStd[i].nim == search) {
                potition = i;
                break;
            }
        }
        return potition;
    }
}

```

```

    public void showPotition(int x, int pos) {
        if (pos != 1) {
            System.out.println("Data : " + x + " is found in
                               ↪ index-" + pos);
        } else {
            System.out.println("Data : " + x + "is not found");
        }
    }

    public void showData(int x, int pos) {
        if (pos != 1) {
            System.out.println("NIM \t : " + x);
            System.out.println("Name \t : " + listStd[pos].name);
            System.out.println("Age \t : " + listStd[pos].age);
            System.out.println("IPK \t : " + listStd[pos].gpa);
        } else {
            System.out.println("Data " + x + " is not found");
        }
    }
}

package week7;

import java.util.Scanner;

public class StudentsMain {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        Scanner sl = new Scanner(System.in);

        SearchStudent data = new SearchStudent();
        int amountStudent = 5;

        System.out.println("-----");
        System.out.println("Input student data accordingly from
                               ↪ samllest NIM");
        for (int i = 0; i < amountStudent; i++) {
            System.out.println("-----");
            System.out.print("NIM\t: ");
            int nim = s.nextInt();
            System.out.print("Name\t: ");
            String name = sl.nextLine();

```

```

        System.out.print("Age\t: ");
        int age = s.nextInt();
        System.out.print("GPA\t: ");
        double gpa = s.nextDouble();

        Students std = new Students(nim, age, name, gpa);
        data.add(std);
    }

    System.out.println("-----");
    System.out.println("Entire Student Data");
    data.display();

    System.out.println("-----");
    System.out.println("-----");
    System.out.print("Search student by NIM: ");
    int search = s.nextInt();
    System.out.println("Using Sequential Search");
    int potition = data.findSeqSearch(search);

    data.showPotition(search, potition);

    data.showData(search, potition);

    s.close();
    sl.close();
}
}

```

1.2.2 Result

```

1 PS D:\Kuliah> d:; cd 'd:\Kuliah'; & 'C:\Program
   ↳ Files\Java\jdk-18.0.2.1\bin\java.exe'
   ↳ '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
   ↳ 'C:\Users\ASUS\AppData\Roaming\Code\User\workspaceStorage\ce3fcb236261368a6cbd019
   ↳ 'week7.StudentsMain'
2 -----
3 Input student data accordingly from samllest NIM
4 -----
5 NIM      : 2017
6 Name     : Dewi Lestari
7 Age      : 23

```

```
8 GPA      : 3.5
9 -----
10 NIM      : 2018
11 Name     : Sinta Sanjaya
12 Age      : 22
13 GPA      : 4
14 -----
15 NIM      : 2019
16 Name     : Danang Adi
17 Age      : 22
18 GPA      : 3.7
19 -----
20 NIM      : 2020
21 Name     : Budi Prakarsa
22 Age      : 20
23 GPA      : 2.9
24 -----
25 NIM      : 2021
26 Name     : Vania Siti
27 Age      : 20
28 GPA      : 3
29 -----
30 Entire Student Data
31 NIM =2017
32 Name =Dewi Lestari
33 Age =23
34 GPA =3.5
35 -----
36 NIM =2018
37 Name =Sinta Sanjaya
38 Age =22
39 GPA =4.0
40 -----
41 NIM =2019
42 Name =Danang Adi
43 Age =22
44 GPA =3.7
45 -----
46 NIM =2020
47 Name =Budi Prakarsa
48 Age =20
49 GPA =2.9
```

```

50  -----
51  NIM =2021
52  Name =Vania Siti
53  Age =20
54  GPA =3.0
55  -----
56  -----
57  -----
58  Search student by NIM: 2017
59  Using Sequential Search
60  Data : 2017 is found in index-0
61  NIM      : 2017
62  Name     : Dewi Lestari
63  Age      : 23
64  IPK      : 3.5

```

1.2.3 Question

1. What is the difference of method **displayData** and **displayPosition** in **StudentSearch** class?

Answer:

displayData shows the data being search by index in the list of students. displayPosition shows the index of the searched data in the list.

2. What is the function of break in this following program code?

```

if (listStd[i].nim == search) {
    potition = i;
    break;
}

```

Answer:

breaking the loop once the data being search has been founded.

3. If inserted NIM data is not sorted from smallest to biggest value, will the program encounter an error? Is the result still correct? Why is that?

Answer:

the search algorithm will output the wrong data because the search algorithm search sequentially trough the index.

1.3 Binary Search Method

1.3.1 Steps

1. in step 1.2.1 (Sequential search), create method FindBinarySearch with integer as its data type in class SearchStudent. Then declare the content of method FindBinarySearch with using binary search as its searching algorithm
2. Call method FindBinarySearch from SearchStudent class in StudentsMain. Then call method displayPosition and displayData
3. Run and see the result

1.3.2 Question

1. Show the program code in which runs the divide process

Answer:

```
if (find == listStd[mid].nim) {  
    return mid;  
} else if (listStd[mid].nim > find) {  
    return findBinarySearch(find, left, mid - 1);  
} else {  
    return findBinarySearch(find, mid + 1, right);  
}
```

2. Show the program code in which runs the conquer process

Answer:

```
return mid;  
  
return findBinarySearch(find, left, mid - 1);  
  
return findBinarySearch(find, mid + 1, right);
```

3. If inserted NIM data is not sorted, will the program crash? Why? If inserted NIM data is sorted from largest to smallest value (e.g 20215, 20214 20212, 20211,20210) and element being searched is 20210. How is the result of binary search? does it return the correct one? if not, then change the code so that the binary search executed properly

Answer:

the program will output an error if the data is not sorted. because the program works by comparing the value of the search item with the division of the divide conquer index.

-
4. Modify program above so that the students amount inserted is matched with user input

Answer :