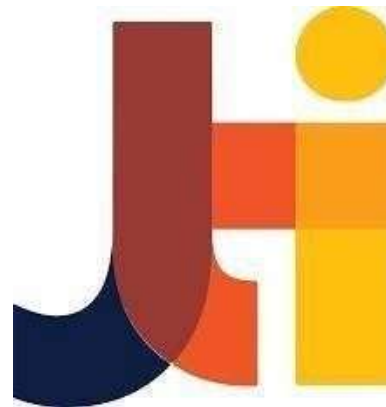


LAPORAN PRAKTIKUM
MATA KULIAH PRAKTIKUM ALGORITMA STRUKTUR DATA

Dosen Pengampu : Triana Fatmawati, S.T, M.T

PERTEMUAN 7 : SEARCHING



Nama : Yonanda Mayla Rusdiaty

NIM : 2341760184

Prodi : D-IV Sistem Informasi Bisnis

JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

2024

6.2 Searching / Pencarian Menggunakan Algoritma Sequential Search

6.2.1 Sequential Search Menggunakan Array

Kode Program :

```
1 public class Sorting29 {
2     public static void sequentialSearch(int[] arr, int key) {
3         for (int i = 0; i < arr.length; i++) {
4             if (arr[i] == key) {
5                 System.out.println("Data ditemukan pada indeks ke - " + i);
6             }
7         }
8         System.out.println("Data tidak ditemukan");
9     }
10
11     public static void main(String[] args) {
12         int[] daftarNilai = {10, 5, 20, 15, 80, 45};
13         sequentialSearch(daftarNilai, 5);
14     }
15 }
```

Output :

```
Data ditemukan pada indeks ke - 1
Data tidak ditemukan
PS D:\KULIAH\college\smt 2\29_yonanda_asd\P7>
```

6.2.2 Sequential Search Menggunakan Array of Object

Kode Program clas Mahasiswa29 :

```
1  public class Mahasiswa29 {
2      int nim;
3      String nama;
4      int umur;
5      double ipk;
6
7      Mahasiswa29(int ni, String n, int u, double i) {
8          nim = ni;
9          nama = n;
10         umur = u;
11         ipk = i;
12     }
13
14     void tampil() {
15         System.out.println("Nim : " + nim);
16         System.out.println("Nama : " + nama);
17         System.out.println("Umur : " + umur);
18         System.out.println("IPK : " + ipk);
19     }
20 }
21
```

Kode Program class PencarianMhs29 :

```
1 public class PencarianMhs29 {
2     Mahasiswa29 listMhs[] = new Mahasiswa29[5];
3     int idx;
4
5     void tambah(Mahasiswa29 m) {
6         if (idx < listMhs.length) {
7             listMhs[idx] = m;
8             idx++;
9         } else {
10             System.out.println("Data sudah penuh!!");
11         }
12     }
13
14     void tampil() {
15         for (Mahasiswa29 m : listMhs) {
16             m.tampil();
17             System.out.println("-----");
18         }
19     }
20
21     public int FindSeqSearch(int cari) {
22         int posisi = -1;
23         for (int j = 0; j < listMhs.length; j++) {
24             if (listMhs[j].nim == cari) {
25                 posisi = j;
26                 break;
27             }
28         }
29         return posisi;
30     }
31
32     public void TampilPosisi(int x, int pos) {
33         if (pos != -1) {
34             System.out.println("Data : " + x + "ditemukan pada indeks : " + pos);
35         } else {
36             System.out.println("Data " + x + "tidak ditemukan");
37         }
38     }
39
40     public void TampilData(int x, int pos) {
41         if (pos != -1) {
42             System.out.println("Nim\t: " + x);
43             System.out.println("Nama\t: " + listMhs[pos].nama);
44             System.out.println("Umur\t: " + listMhs[pos].umur);
45             System.out.println("IPK\t: " + listMhs[pos].ipk);
46         } else {
47             System.out.println("Data " + x + "tidak ditemukan");
48         }
49     }
50 }
51
```

Kode Program class MainMahasiswa29 :

```
1  import java.util.Scanner;
2
3  public class MainMahasiswa29 {
4      public static void main(String[] args) {
5          Scanner s = new Scanner(System.in); // scanner untuk int
6          Scanner s1 = new Scanner(System.in); // scanner untuk string
7
8          PencarianMhs29 data = new PencarianMhs29();
9          int jumMhs = 5;
10
11         System.out.println("=====");
12         System.out.println("Masukkan data mahasiswa secaraurut dari NIM terkecil : ");
13         for (int i = 0; i < jumMhs; i++) {
14             System.out.println("=====");
15             System.out.print("Nim\t : ");
16             int nim = s.nextInt();
17             System.out.print("Nama\t : ");
18             String nama = s1.nextLine();
19             System.out.print("Umur\t : ");
20             int umur = s.nextInt();
21             System.out.print("IPK\t : ");
22             double ipk = s.nextDouble();
23
24             Mahasiswa29 m = new Mahasiswa29(nim, nama, umur, ipk);
25             data.tambah(m);
26         }
27
28         System.out.println("=====");
29         System.out.println("Data keseluruhan mahasiswa : ");
30         data.tampil();
31
32         System.out.println("-----");
33         System.out.println("-----");
34         System.out.println("Pencarian Data : ");
35         System.out.println("Masukkan NIM mahasiswa yang dicari : ");
36         System.out.print("NIM : ");
37         int cari = s.nextInt();
38         System.out.println("Menggunakan Sequential Search");
39         int posisi = data.FindSeqSearch(cari);
40
41         data.TampilPosisi(cari, posisi);
42         data.TampilData(cari, posisi);
43     }
44 }
45
```

Output :

```
Masukkan data mahasiswa secara urut dari NIM terkecil :
=====
Nim      : 2017
Nama     : Dewi Iestari
Umur     : 23
IPK      : 3.5
=====
Nim      : 2018
Nama     : Sinta Sanjaya
Umur     : 22
IPK      : 4
=====
Nim      : 2019
Nama     : Danang Adi
Umur     : 22
IPK      : 3.7
=====
Nim      : 2020
Nama     : Budi Prakarsa
Umur     : 20
IPK      : 2.9
=====
Nim      : 2021
Nama     : Vania Siti
Umur     : 20
IPK      : 3.0
=====
Data keseluruhan mahasiswa :
Nim : 2017
Nama : Dewi Iestari
Umur : 23
IPK : 3.5
=====
Nim : 2018
Nama : Sinta Sanjaya
Umur : 22
IPK : 4.0
=====

Nim : 2018
Nama : Sinta Sanjaya
Umur : 22
IPK : 4.0
=====
Nim : 2019
Nama : Danang Adi
Umur : 22
IPK : 3.7
=====
Nim : 2020
Nama : Budi Prakarsa
Umur : 20
IPK : 2.9
=====
Nim : 2021
Nama : Vania Siti
Umur : 20
IPK : 3.0
=====

Pencarian Data :
Masukkan NIM mahasiswa yang dicari :
NIM : 2018
Menggunakan Sequential Search
Data : 2018 ditemukan pada indeks : 1
Nim      : 2018
Nama     : Sinta Sanjaya
Umur     : 22
IPK      : 4.0
PS D:\KULIAH\college\smt 2\29_yonanda_asd\p7>
```

Pertanyaan :

1. Lakukan perubahan array daftarNilai pada fungsi main()

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

Jawab : Berikut adalah hasil modifikasinya :

a) Kode program :

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

b) Output :

```

-----
Pencarian Data :
Masukkan NIM mahasiswa yang dicari :
NIM : 2018
Menggunakan Sequential Search
Data : 2018ditemukan pada indeks : 1
Nim      : 2018
Nama     : Sinta Sanjaya
Umur     : 22
IPK      : 4.0
=====
Menggunakan Binary Search
Data : 2018ditemukan pada indeks : 1
Nim      : 2018
Nama     : Sinta Sanjaya
Umur     : 22
IPK      : 4.0
PS D:\KULIAH\college\smt 2\29_yonanda_asd\P7>

```

2. Jelaskan perbedaan metod TampilData dan Tampilposisi pada class PencarianMhs!

Jawab : Perbedaanya adalah sebagai berikut :

- TampilData → untuk menampilkan data mahasiswa dari masukan user berupa nim, nama, umur, dan ipk, jika posisi ditemukan
- TampilPosisi → digunakan untuk mngecek posisi data yang dicari pada indeks ke berapa, jika ada akan ada ditampilkan data indeks berapa nya. Namun, jika tidak data yang dicari tidak ada pada indeks, maka akan mengeprintkan data tidak ditemukan

3. Jelaskan fungsi break pada kode program dibawah ini!

```

if (listMHs[j].nim==cari) {
    posisi = j;
    break;
}

```

Jawab : Break tersebut berfungsi untuk menghentikan looping jika data yang dicari berhasil ditemukan. Jika tidak ada break, maka looping akan terus menerus terjadi

4. Jika Data Nim yang dimasukkan tidak terurut dari kecil ke besar. Apakah program masih dapat berjalan? Apakah hasil yang dikeluarkan benar? Mengapa demikian!

Jawab : Program masih akan tetap berjalan meskipun data NIM yang dimasukkan tidak terurut dari kecil ke besar. Namun, hasil yang dikeluarkan mungkin tidak benar untuk metode pencarian Binary Search (karena membutuhkan data yang terurut), sementara hasilnya tetap benar untuk metode

pencarian Sequential Search karena dia tidak membutuhkan data yang terurut.

6.3 Searching / Pencarian Menggunakan Binary Search

6.3.1 Langkah-langkah Percobaan Binary Search menggunakan Array

Kode program :

```
1 public class Sorting29 {
2     public static void sequentialSearch(int[] arr, int key) {
3         for (int i = 0; i < arr.length; i++) {
4             if (arr[i] == key) {
5                 System.out.println("Data ditemukan pada indeks ke - " + i);
6             }
7         }
8         System.out.println("Data tidak ditemukan");
9     }
10
11     public static int binarySearchAsc(int[] arr, int key) {
12         int start = 0, end = arr.length - 1;
13
14         while (start <= end) {
15             int mid = start + (end - start) / 2;
16
17             if (arr[mid] == key) {
18                 return mid;
19             }
20
21             if (arr[mid] < key) {
22                 start = mid + 1;
23             } else {
24                 end = mid - 1;
25             }
26         }
27         return -1;
28     }
29
30     public static void main(String[] args) {
31         int[] daftarNilai = {10, 5, 20, 15, 80, 45};
32         sequentialSearch(daftarNilai, 5);
33
34         int[] sortedNilai = {5, 5, 10, 20, 30, 40, 50};
35         int index = binarySearchAsc(daftarNilai, 5);
36
37         if (index != -1) {
38             System.out.println("Data ditemukan pada indeks ke - " + index);
39         } else {
40             System.out.println("Data tidak ditemukan");
41         }
42     }
43 }
```

Output :

```
-----
Pencarian Data :
Masukkan NIM mahasiswa yang dicari :
NIM : 2018
Menggunakan Sequential Search
Data : 2018ditemukan pada indeks : 1
Nim      : 2018
Nama     : Sintya Sanjaya
Umur     : 22
IPK      : 4.0
=====
Menggunakan Binary Search
Data : 2018ditemukan pada indeks : 1
Nim      : 2018
Nama     : Sintya Sanjaya
Umur     : 22
IPK      : 4.0
PS D:\KULIAH\college\smt 2\29_yonanda_asd\P7> |
```

Pertanyaan :

1. Tunjukkan pada kode program yang mana proses divide dijalankan!

Jawab :

```
if (right >= left) {
    mid = (left + right) / 2 ;
```

Value dari data kanan dan kiri digunakan untuk menghitung nilai tengah dari rentang pencarian. Dimana hal tersebut, merupakan langkah pembagian yang dilakukan oleh algoritma Binary Search (membagi 2 tiap iterasi)

2. Tunjukkan pada kode program yang mana proses conquer dijalankan!

Jawab :

```
if (cari == listMhs[mid].nim) {
    return(mid);
```

Ketika nilai yang dicari (cari) sama dengan nilai NIM di tengah (listMhs[mid].nim), algoritma menemukan elemen yang dicari dan mengembalikan indeks tengah mid. Ini menunjukkan bahwa proses conquer telah berhasil dan elemen yang dicari telah ditemukan di tengah array.

3. Jika data Nim yang dimasukkan tidak urut. Apakah program masih dapat berjalan? Mengapa demikian!

Jawab : Jika data NIM yang dimasukkan tidak urut, maka hasil dari pencarian yang dilakukan oleh algoritma Binary Search tidak dapat diandalkan. Hal ini disebabkan karena algoritma Binary Search mengasumsikan bahwa data yang akan dicari sudah dalam keadaan terurut. Jika data tidak terurut, maka algoritma tersebut tidak akan dapat menemukan hasil yang konsisten.

4. Jika Nim yang dimasukkan dari NIM terbesar ke terkecil (missal : 20215, 20214, 20212, 20211, 20210) dan elemen yang dicari adalah 20210. Bagaimana hasil dari binary search? Apakah sesuai? Jika tidak sesuai maka ubahlah kode program binary seach agar hasilnya sesuai

Jawab : Jika data NIM dimasukkan dari terbesar ke terkecil (misalnya: 20215, 20214, 20212, 20211,

20210) dan elemen yang dicari adalah 20210, hasil dari pencarian menggunakan algoritma Binary Search tidak akan sesuai. Hal ini disebabkan karena algoritma Binary Search mengasumsikan bahwa data dalam array telah diurutkan secara menaik (ascending).

5. Modifikasilah program diatas yang mana jumlah mahasiswa yang di inputkan sesuai dengan masukan dari keyboard.

Jawab :

Output :

```
=====
Data keseluruhan mahasiswa :
Nim : 2022
Nama : Kresna
Umur : 20
IPK : 4.0
=====
Nim : 2023
Nama : Yonanda
Umur : 18
IPK : 3.9
=====
-----
Pencarian Data :
Masukkan NIM mahasiswa yang dicari :
NIM : 2023
Menggunakan Sequential Search
Data : 2023 ditemukan pada indeks : 1
Nim      : 2023
Nama     : Yonanda
Umur     : 18
IPK      : 3.9
=====
Menggunakan Binary Search
Data : 2023 ditemukan pada indeks : 1
Nim      : 2023
Nama     : Yonanda
Umur     : 18
IPK      : 3.9
PS D:\KULIAH\college\smt 2\29_yonanda_asd\p7> 
```

Kode program :

```
1  import java.util.Scanner;
2
3  public class MainMahasiswa29 {
4      public static void main(String[] args) {
5          Scanner s = new Scanner(System.in); // scanner untuk int
6          Scanner sl = new Scanner(System.in); // scanner untuk string
7
8          System.out.print("Masukkan jumlah mahasiswa: ");
9          int jumMhs = s.nextInt();
10
11          // Membuat objek PencarianMhs29 dengan konstruktor yang sesuai
12          PencarianMhs29 data = new PencarianMhs29(jumMhs);
13
14          System.out.println("=====");
15          System.out.println("Masukkan data mahasiswa secaraurut dari NIM terkecil : ");
16          for (int i = 0; i < jumMhs; i++) {
17              System.out.println("=====");
18              System.out.print("Nim\t : ");
19              int nim = s.nextInt();
20              System.out.print("Nama\t : ");
21              String nama = sl.nextLine();
22              System.out.print("Umur\t : ");
23              int umur = s.nextInt();
24              System.out.print("IPK\t : ");
25              double ipk = s.nextDouble();
26
27              Mahasiswa29 m = new Mahasiswa29(nim, nama, umur, ipk);
28              data.tambah(m);
29          }
30
31          System.out.println("=====");
32          System.out.println("Data keseluruhan mahasiswa : ");
33          data.tampil();
34
35          System.out.println("=====");
36          System.out.println("Pencarian Data : ");
37          System.out.println("Masukkan NIM mahasiswa yang dicari : ");
38          System.out.print("NIM : ");
39          int cari = s.nextInt();
40          System.out.println("Menggunakan Sequential Search");
41          int posisi = data.FindSeqSearch(cari);
42
43          data.TampilPosisi(cari, posisi);
44          data.TampilData(cari, posisi);
45
46          System.out.println("=====");
47          System.out.println("Menggunakan Binary Search");
48          posisi = data.FindBinarySearch(cari, 0, jumMhs - 1);
49          data.TampilPosisi(cari, posisi);
50          data.TampilData(cari, posisi);
51      }
52  }
53
```

6.4 Percobaan Pengayaan Divide and Conquer

6.4.1. Langkah-langkah Percobaan Merge Sort

Kode Program class MergeSorting29 :

```
1  public class MergeSorting29 {
2      public void mergeSort(int[] data) {
3          sort(data, 0, data.length - 1);
4      }
5
6
7      public void merge(int[] data, int left, int middle, int right) {
8          int[] temp = new int[data.length];
9          for (int i = left; i <= right; i++) {
10             temp[i] = data[i];
11          }
12
13          int a = left;
14          int b = middle + 1;
15          int c = left;
16
17          // membandingkan setiap bagian
18          while (a <= middle && b <= right) {
19              if (temp[a] <= temp[b]) {
20                  data[c] = temp[a];
21                  a++;
22              } else {
23                  data[c] = temp[b];
24                  b++;
25              }
26              c++;
27          }
28          int s = middle - a;
29          for (int i = 0; i <= s; i++) {
30              data[c + i] = temp[a + i];
31          }
32      }
33
34      // membagi menjadi 2 bagian dan dibagi kembali hingga tidak dapat dibagi kembali
35      private void sort(int[] data, int left, int right) {
36          if (left < right) {
37              int middle = (left + right) / 2;
38              sort(data, left, middle);
39              sort(data, middle + 1, right);
40              merge(data, left, middle, right);
41          }
42      }
43      public void printArray(int arr[]) {
44          int n = arr.length;
45          for (int i = 0; i < n; i++) {
46              System.out.print(arr[i] + " ");
47          }
48          System.out.println();
49      }
50  }
51
```

Kode program class SortMain29 :

```
1 public class SortMain29 {  
2     public static void main(String[] args) {  
3         int data[] = {10, 40, 30, 50, 70, 20, 100, 90};  
4         System.out.println("Sorting dengan merge sort");  
5         MergeSorting29 mSort = new MergeSorting29();  
6         System.out.println("Data Awal");  
7         mSort.printArray(data);  
8         mSort.mergeSort(data);  
9         System.out.println("Setelah diurutkan");  
10        mSort.printArray(data);  
11    }  
12 }
```

Output :

Sorting dengan merge sort

Data Awal

10 40 30 50 70 20 100 90

Setelah diurutkan

10 20 30 40 50 70 90 100

PS D:\KULIAH\college\smt 2\29_yonanda_asd\P7> █

LATIHAN PRAKTIKUM :

Modifikasi percobaan searching diatas yang menggunakan Searching array of object dengan ketentuan berikut ini - Pencarian dilakukan berdasarkan Nama Mahasiswa (gunakan Algoritma binary Search) - Buat aturan untuk mendeteksi hasil pencarian lebih dari 1 hasil dalam bentuk kalimat peringatan!

JAWABAN :

Berikut adalah hasil modifikasinya :

Code Program :

```

1 // Header file for math.h
2 #include <math.h>
3
4 // Function to calculate the sum of squares of first n natural numbers
5 int sum_of_squares(int n)
6 {
7     int sum = 0;
8     for (int i = 1; i <= n; i++)
9     {
10         sum += i * i;
11     }
12     return sum;
13 }
14
15 // Function to calculate the sum of cubes of first n natural numbers
16 int sum_of_cubes(int n)
17 {
18     int sum = 0;
19     for (int i = 1; i <= n; i++)
20     {
21         sum += i * i * i;
22     }
23     return sum;
24 }
25
26 // Function to calculate the sum of squares of first n even numbers
27 int sum_of_squares_even(int n)
28 {
29     int sum = 0;
30     for (int i = 1; i <= n; i++)
31     {
32         sum += (2 * i) * (2 * i);
33     }
34     return sum;
35 }
36
37 // Function to calculate the sum of cubes of first n even numbers
38 int sum_of_cubes_even(int n)
39 {
40     int sum = 0;
41     for (int i = 1; i <= n; i++)
42     {
43         sum += (2 * i) * (2 * i) * (2 * i);
44     }
45     return sum;
46 }
47
48 // Function to calculate the sum of squares of first n odd numbers
49 int sum_of_squares_odd(int n)
50 {
51     int sum = 0;
52     for (int i = 1; i <= n; i++)
53     {
54         sum += (2 * i - 1) * (2 * i - 1);
55     }
56     return sum;
57 }
58
59 // Function to calculate the sum of cubes of first n odd numbers
60 int sum_of_cubes_odd(int n)
61 {
62     int sum = 0;
63     for (int i = 1; i <= n; i++)
64     {
65         sum += (2 * i - 1) * (2 * i - 1) * (2 * i - 1);
66     }
67     return sum;
68 }
69
70 // Function to calculate the sum of squares of first n natural numbers using formula
71 int sum_of_squares_formula(int n)
72 {
73     return n * (n + 1) * (2 * n + 1) / 6;
74 }
75
76 // Function to calculate the sum of cubes of first n natural numbers using formula
77 int sum_of_cubes_formula(int n)
78 {
79     return n * n * (n + 1) * (n + 1) / 4;
80 }
81
82 // Function to calculate the sum of squares of first n even numbers using formula
83 int sum_of_squares_even_formula(int n)
84 {
85     return 4 * n * (n + 1) * (2 * n + 1) / 3;
86 }
87
88 // Function to calculate the sum of cubes of first n even numbers using formula
89 int sum_of_cubes_even_formula(int n)
90 {
91     return 16 * n * n * (n + 1) * (n + 1) / 3;
92 }
93
94 // Function to calculate the sum of squares of first n odd numbers using formula
95 int sum_of_squares_odd_formula(int n)
96 {
97     return n * (2 * n - 1) * (2 * n + 1) / 3;
98 }
99
100 // Function to calculate the sum of cubes of first n odd numbers using formula
101 int sum_of_cubes_odd_formula(int n)
102 {
103     return n * n * (2 * n - 1) * (2 * n + 1) / 3;
104 }
105
106 // Main function
107 int main()
108 {
109     int n;
110     printf("Enter the value of n: ");
111     scanf("%d", &n);
112
113     // Calculate the sum of squares of first n natural numbers
114     int sum_squares = sum_of_squares(n);
115     printf("Sum of squares of first %d natural numbers: %d\n", n, sum_squares);
116
117     // Calculate the sum of cubes of first n natural numbers
118     int sum_cubes = sum_of_cubes(n);
119     printf("Sum of cubes of first %d natural numbers: %d\n", n, sum_cubes);
120
121     // Calculate the sum of squares of first n even numbers
122     int sum_squares_even = sum_of_squares_even(n);
123     printf("Sum of squares of first %d even numbers: %d\n", n, sum_squares_even);
124
125     // Calculate the sum of cubes of first n even numbers
126     int sum_cubes_even = sum_of_cubes_even(n);
127     printf("Sum of cubes of first %d even numbers: %d\n", n, sum_cubes_even);
128
129     // Calculate the sum of squares of first n odd numbers
130     int sum_squares_odd = sum_of_squares_odd(n);
131     printf("Sum of squares of first %d odd numbers: %d\n", n, sum_squares_odd);
132
133     // Calculate the sum of cubes of first n odd numbers
134     int sum_cubes_odd = sum_of_cubes_odd(n);
135     printf("Sum of cubes of first %d odd numbers: %d\n", n, sum_cubes_odd);
136
137     // Calculate the sum of squares of first n natural numbers using formula
138     int sum_squares_formula = sum_of_squares_formula(n);
139     printf("Sum of squares of first %d natural numbers (formula): %d\n", n, sum_squares_formula);
140
141     // Calculate the sum of cubes of first n natural numbers using formula
142     int sum_cubes_formula = sum_of_cubes_formula(n);
143     printf("Sum of cubes of first %d natural numbers (formula): %d\n", n, sum_cubes_formula);
144
145     // Calculate the sum of squares of first n even numbers using formula
146     int sum_squares_even_formula = sum_of_squares_even_formula(n);
147     printf("Sum of squares of first %d even numbers (formula): %d\n", n, sum_squares_even_formula);
148
149     // Calculate the sum of cubes of first n even numbers using formula
150     int sum_cubes_even_formula = sum_of_cubes_even_formula(n);
151     printf("Sum of cubes of first %d even numbers (formula): %d\n", n, sum_cubes_even_formula);
152
153     // Calculate the sum of squares of first n odd numbers using formula
154     int sum_squares_odd_formula = sum_of_squares_odd_formula(n);
155     printf("Sum of squares of first %d odd numbers (formula): %d\n", n, sum_squares_odd_formula);
156
157     // Calculate the sum of cubes of first n odd numbers using formula
158     int sum_cubes_odd_formula = sum_of_cubes_odd_formula(n);
159     printf("Sum of cubes of first %d odd numbers (formula): %d\n", n, sum_cubes_odd_formula);
160
161     return 0;
162 }

```

Output :


```
Masukkan jumlah mahasiswa: 2
=====
Masukkan data mahasiswa secara urut dari NIM terkecil :
=====
Nim      : 2021
Nama     : Kresna
Umur     : 20
IPK      : 4.0
=====
Nim      : 2023
Nama     : Yonanda
Umur     : 18
IPK      : 3.9
=====
Data keseluruhan mahasiswa :
Nim      : 2021
Nama     : Kresna
Umur     : 20
IPK      : 4.0
=====
Nim      : 2023
Nama     : Yonanda
Umur     : 18
IPK      : 3.9
=====
Pencarian Data :
Masukkan Nama mahasiswa yang dicari :
Nama: Kresna
Menggunakan Binary Search
Data dengan nama Kresna ditemukan pada indeks : 0
Nim      : 2021
Nama     : Kresna
Umur     : 20
IPK      : 4.0
PS D:\KULIAH\college\smt_2\29_yonanda_asd\VP7>
```