**LAPORAN TUGAS ALGORITMA DAN STRUKTUR DATA**

**JOBSHEET 10**

`

NAMA : RADITYA RIEFKI

KELAS : TI 1E

ABSEN : 23

**JOBSHEET VII SEARCHING**

7.2. Searching / Pencarian Menggunakan Agoritma Sequential Search

1.Kode Program

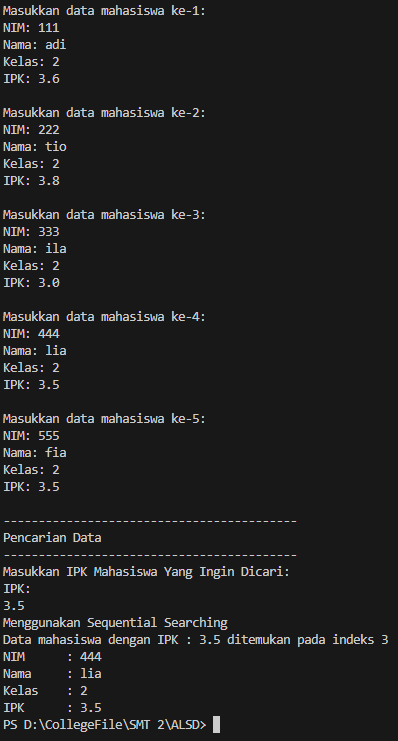
Mahasiswa Berprestasi

|  |
| --- |
| int sequentialSearching(double cari){  int posisi = -1;  for (int j = 0; j < listMhs.length; j++) {  if (listMhs[j].ipk==cari) {  posisi = j;  break;  }  }  return posisi;  }  void tampilPosisi(double x, int pos){  if (pos!=-1) {  System.out.println("Data mahasiswa dengan IPK : " + x + " ditemukan pada indeks " + (pos));  }else{  System.out.println("Data " + x + "tidak ditemukan");  }  }  void tampilDataSearch(double x, int pos){  if (pos !=-1) {  System.out.println("NIM\t : " + listMhs[pos].nim);  System.out.println("Nama\t : " + listMhs[pos].nama);  System.out.println("Kelas\t : " + listMhs[pos].kelas);  System.out.println("IPK\t : " + x);  }else{  System.out.println("Data mahasiswa dengan IPK : " + x + "tidak ditemukan");  } |

Mahasiswa Demo

|  |
| --- |
| package jobsheet7;  import java.util.Scanner;  public class MahasiswaDemo23 {  public static void main(String[] args) {  Scanner sc = new Scanner(System.in);    int jumMhs = 5;    MahasiswaBerprestasi23 list = new MahasiswaBerprestasi23(jumMhs);    for (int i = 0; i < jumMhs; i++) {  System.out.println("Masukkan data mahasiswa ke-" + (i + 1) + ":");  System.out.print("NIM: ");  String nim = sc.nextLine();  System.out.print("Nama: ");  String nama = sc.nextLine();  System.out.print("Kelas: ");  String kelas = sc.nextLine();  System.out.print("IPK: ");  double ipk = sc.nextDouble();  sc.nextLine();  Mahasiswa23 m = new Mahasiswa23(nim, nama, kelas, ipk);  list.tambah(m);  System.out.println();  }  System.out.println("------------------------------------------");  System.out.println("Pencarian Data");  System.out.println("------------------------------------------");  System.out.println("Masukkan IPK Mahasiswa Yang Ingin Dicari: ");  System.out.println("IPK: ");  double cari = sc.nextDouble();  System.out.println("Menggunakan Sequential Searching");  double posisi = list.sequentialSearching(cari);  int pss = (int) posisi;  list.tampilPosisi(cari, pss);  list.tampilDataSearch(cari, pss);  }  } |

2. Output



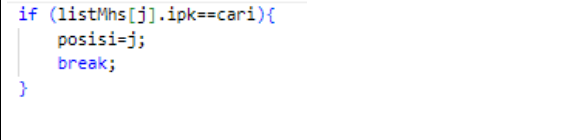
**7.2.3. Pertanyaan**

1. Jelaskan perbedaan metod tampilDataSearch dan tampilPosisi pada class MahasiswaBerprestasi!

- method tampilPosisi digunakan untuk memberi tahu posisi indeks ke berapa dalam data array

Method tampilDataSearch digunakan untuk menampiljan informasi lengkap tentang mahasiswa yang dicari berdasarkan IPK

2. Jelaskan fungsi break pada kode program dibawah ini!



- Fungsi break pada kode diatas digunakan jika kondisi listMhs[j].ipk sama dengan input cari maka akan berhenti

**7.3. Searching / Pencarian Menggunakan Binary Search**

1. Kode Program

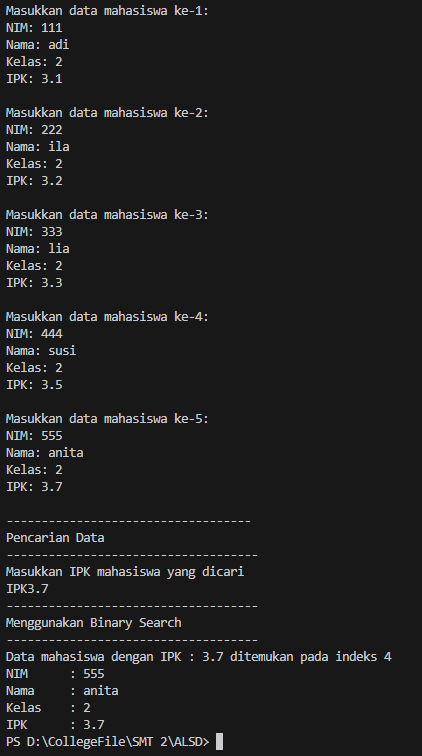
Mahasiswa Berprestasi

|  |
| --- |
| int findBinarySearch(double cari, int left, int right){  int mid;  if (right >= left) {  mid = left + (right - left) / 2;  if (cari == listMhs[mid].ipk) {  return (mid);  }  else if (listMhs[mid].ipk > cari) {  return findBinarySearch(cari, left, mid -1);  }  else{  return findBinarySearch(cari, mid + 1, right);  }  }  return -1;  } |

Mahasiswa Demo

|  |
| --- |
| System.out.println("-----------------------------------");  System.out.println("Pencarian Data");  System.out.println("------------------------------------");  System.out.println("Masukkan IPK mahasiswa yang dicari");  System.out.print("IPK: ");  double cari = sc.nextDouble();  sc.nextLine();  System.out.println("------------------------------------");  System.out.println("Menggunakan Binary Search");  System.out.println("------------------------------------");  double posisi2 = list.findBinarySearch(cari, 0, jumMhs-1);  int pss2 = (int) posisi2;  list.tampilPosisi(cari, pss2);  list.tampilDataSearch(cari, pss2); |

2. Output



**7.3.3. Pertanyaan**

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

- mid = left + (right – left );

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

- return findBinarySearch( cari, left, mid -1);

Dan

return findBinarySearch(cari, mid + 1, right);

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

- binary search hanya berfungsi jika data yang ingin dicari sudah urut jika data belum urut maka harus diurutkan terlebih dahulu

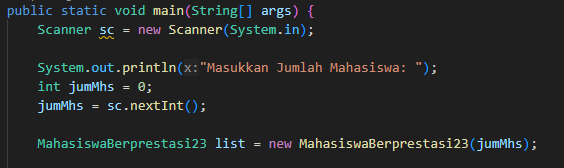
4. Jika IPK yang dimasukkan dari IPK terbesar ke terkecil (missal : 3.8, 3.7, 3.5, 3.4, 3.2) dan elemen yang dicari adalah 3.2. Bagaimana hasil dari binary search? Apakah sesuai? Jika tidak sesuai maka ubahlah kode program binary seach agar hasilnya sesuai

|  |
| --- |
| int findBinarySearch(double cari, int left, int right){  int mid;  if (right >= left) {  mid = left + (right - left) / 2;  if (cari == listMhs[mid].ipk) {  return (mid);  }  else if (listMhs[mid].ipk < cari) {  return findBinarySearch(cari, left, mid -1);  }  else{  return findBinarySearch(cari, mid + 1, right);  }  }  return -1;  } |

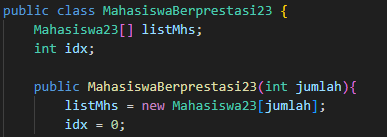
Mengubah kondisi dari (listMhs[mid].ipk > cari) menjadi (listMhs[mid].ipk < cari)

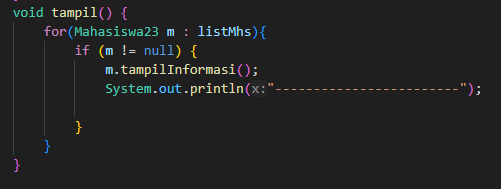
5. Modifikasilah program diatas yang mana jumlah mahasiswa yang di inputkan sesuai dengan masukan dari keyboard.

**Penambahan kode pada program class MahasiswaDemo23**

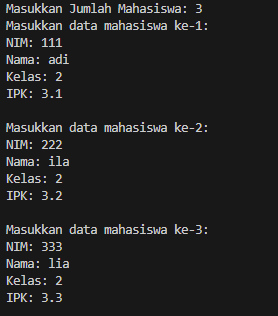
****

**Penambahan kode program pada class MahasiswaBerprestasi23**

****

****

**OUTPUT**

****

**7.5. Latihan Praktikum**

1. Pada Latihan praktikum pertemuan sebelumnya pada Jobsheet 6 yang terdapat 3 class yaitu Dosen, DataDosen , dan DosenDemo, tambahkan method:

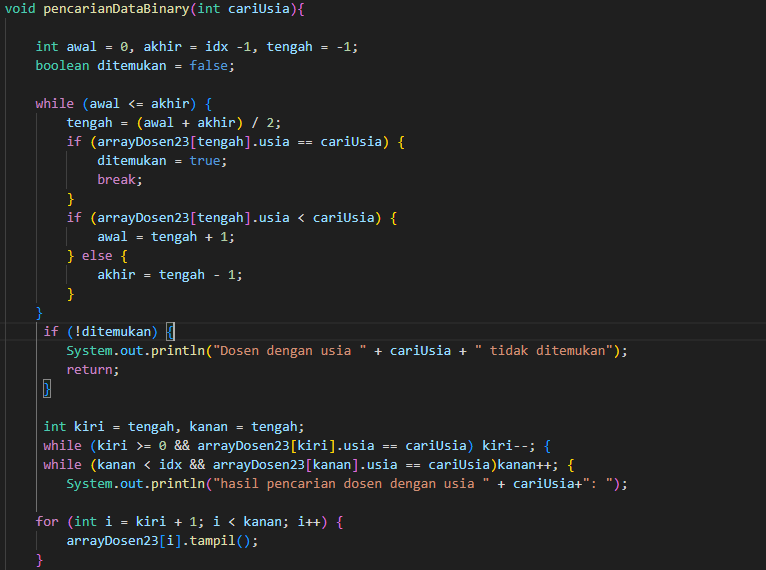
a. PencarianDataSequential : digunakan untuk mencari data dosen berdasarkan nama dengan algoritma sequential search.

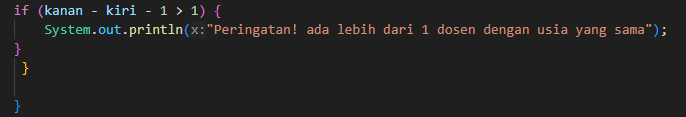
b. PencarianDataBinary : digunakan untuk mencari data dosen berdasarkan usia dengan algoritma Binary Search.

c. Buat aturan untuk mendeteksi hasil pencarian lebih dari 1 hasil dalam bentuk kalimat peringatan! Pastikan algoritma yang diterapkan sesuai dengan kasus yang diberikan!

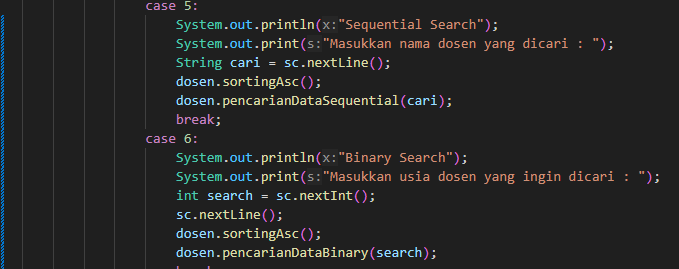
**DataDosen23**



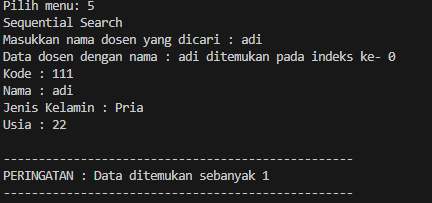


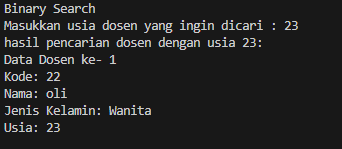


**DosenMain**

****

**OUTPUT**

****

****