Laporan Hasil Praktikum

Algoritma Dan Struktur Data

Jobsheet 7



Angel Chelssa Leoniy Eka Permatasari

244107020202

1E

Program Studi Teknologi Informasi

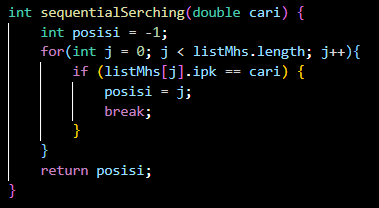
Jurusan Teknik Informatika

POLINEMA

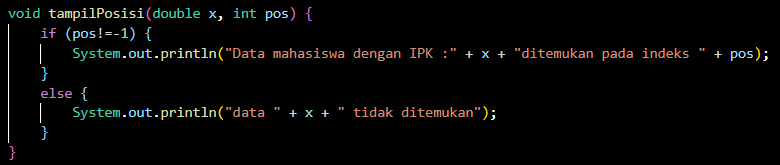
2025

**7.2.1. Langkah-langkah Percobaan Sequential Search**

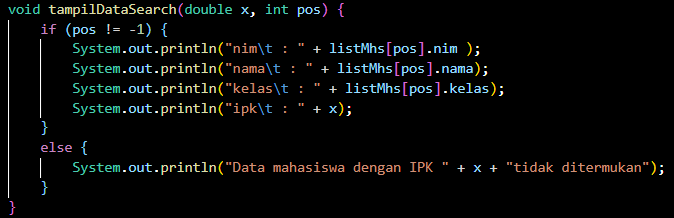
1. Pada pertemuan Jobsheet 7 ini akan menggunakan class Mahasiswa, MahasiswaBerprestasi, dan MahasiswaDemo pada pertemuan Jobsheet 6 sebelumnya
2. Buat folder baru bernama Jobsheet7 di dalam repository Praktikum ASD, kemudian buka ketiga class dari Jobsheet 6 tersebut dan copy ke folder Jobsheet 7
3. Tambahkan method sequentialSearching bertipe integer dengan parameter cari bertipe double pada class MahasiswaBerprestasi. Kemudian Deklarasikan isi method sequentialSearching dengan algoritma pencarian data menggunakan teknik sequential searching.



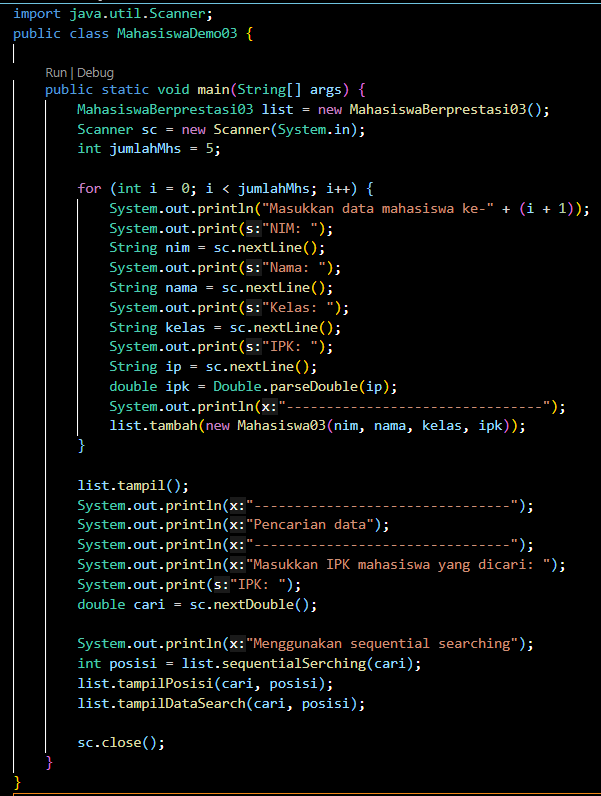
1. Buatlah method tampilPoisisi bertipe void dan Deklarasikan isi dari method tampilPoisisi pada class MahasiswaBerprestasi.



1. Pada class MahasiswaBerprestasi, buatlah method tampilDataSearch bertipe void dan Deklarasikan isi dari method tampilDataSearch.



1. Pada class MahasiswaDemo , tambahkan kode program berikut ini untuk melakukan pencarian data dengan algoritma sequential searching.



**Kode program**

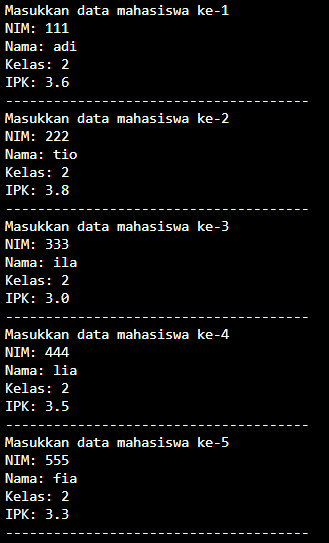
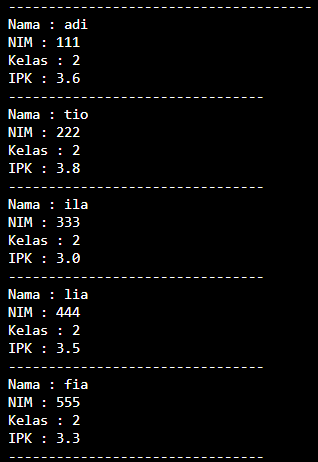
**Kode program MahasiswaBerprestasi03.java**

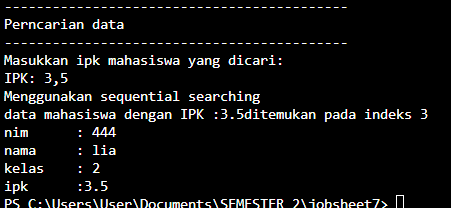
|  |
| --- |
| public class MahasiswaBerprestasi03 {      Mahasiswa03[] listMhs = new Mahasiswa03[5];      int idx;      void tambah(Mahasiswa03 m) {          if (idx < listMhs.length) {              listMhs[idx] = m;              idx++;          } else {              System.out.println("Data sudah penuh!");          }      }      void tampil() {          for (Mahasiswa03 m : listMhs) {              m.tampilInformasi();              System.out.println("--------------------------------");          }      }      void bubbleSort() {          for (int i = 0; i < listMhs.length - 1; i++) {              for (int j = 1; j < listMhs.length - i; j++) {                  if (listMhs[j].ipk > listMhs[j - 1].ipk) {                      Mahasiswa03 tmp = listMhs[j];                      listMhs[j] = listMhs[j - 1];                      listMhs[j - 1] = tmp;                  }              }          }      }      void selectionSort() {          for (int i = 0; i < listMhs.length - 1; i++) {              int idxMin = i; // ✅ diperbaiki              for (int j = i + 1; j < listMhs.length; j++) {                  if (listMhs[j].ipk < listMhs[idxMin].ipk) {                      idxMin = j;                  }              }              Mahasiswa03 tmp = listMhs[idxMin];              listMhs[idxMin] = listMhs[i];              listMhs[i] = tmp;          }      }      void insertionSort() {          for (int i = 1; i < listMhs.length; i++) {              Mahasiswa03 temp = listMhs[i];              int j = i;              while (j > 0 && listMhs[j - 1].ipk > temp.ipk) {                  listMhs[j] = listMhs[j - 1];                  j--;              }              listMhs[j] = temp;          }      }      int sequentialSerching(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");          }      }  } |

**Kode program MahasiswaBerprestasi03.java**

|  |
| --- |
| import java.util.Scanner;  public class MahasiswaDemo03 {      public static void main(String[] args) {          MahasiswaBerprestasi03 list = new MahasiswaBerprestasi03();          Scanner sc  = new Scanner(System.in);          int jumMhs=5;          for(int i=0; i<jumMhs; i++){              System.out.println("Masukkan data mahasiswa ke-" + (i + 1));              System.out.println("NIM: ");              String nim = sc.nextLine();              System.out.println("Nama: ");              String nama = sc.nextLine();              System.out.println("Kelas: ");              String kelas = sc.nextLine();              System.out.println("IPK: ");              String ip =sc.nextLine();              Double ipk = Double.parseDouble(ip);              System.out.println("--------------------------------------");              list.tambah(new Mahasiswa03(nim, nama, kelas, ipk));          }            list.tampil();          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();          System.out.println("Menggunakan sequential searching");          double posisi = list.sequentialSerching(cari);          int pss = (int)posisi;          list.tampilPosisi(cari, pss);          list.tampilDataSearch(cari, pss);      }  } |

**Hasil kode**

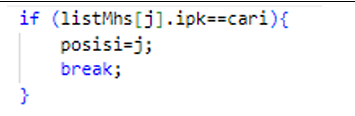


**PERTANYAAN**

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

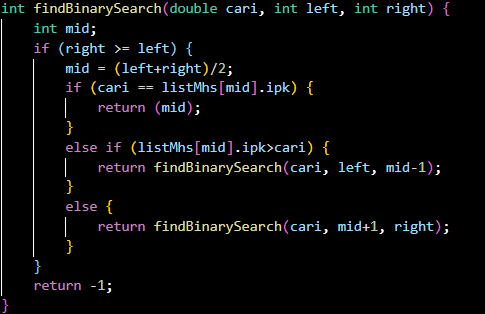
* tampilDataSearch digunakan untuk menampilkan detail lengkap mahasiswa berdasarkan Ipk yang dicari sedangkan tampilPosisi digunakan untuk menampikan posisi(indeks) mahasiswa dalam array listMhs berdasarkan nilai IPK yang dicari

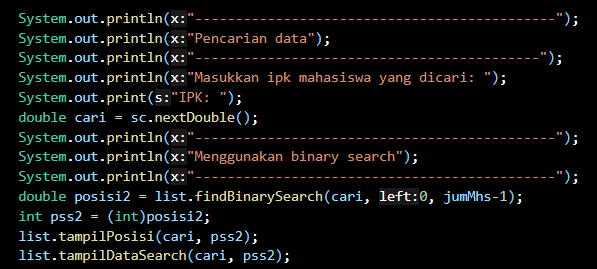
1. Jelaskan fungsi break pada kode program di bawah ini!



* fungsi break pada kode program yang ada ialah menghentikan loop atau perulangan lebih awal jika kondisi if (listMhs[j].ipk == cari ) sudah terpenuhi sehingga pencarian tidak dilanjutkan ke elemen berikutnya

**7.3.1. Langkah-langkah Percobaan Binary Search**

1. Pada percobaan 6.2.1 (sequential search) tambahkan method findBinarySearch bertipe integer pada class MahasiswaBerprestasi. Kemudian Deklarasikan isi method findBinarySearch dengan algoritma pencarian data menggunakan teknik binary searching.
2. Panggil method findBinarySearch terdapat pada class MahasiswaBerprestasi di kelas MahasiswaDemo. Kemudia panggil method tampilPosisi dan tampilDataSearch



**Kode propgram**

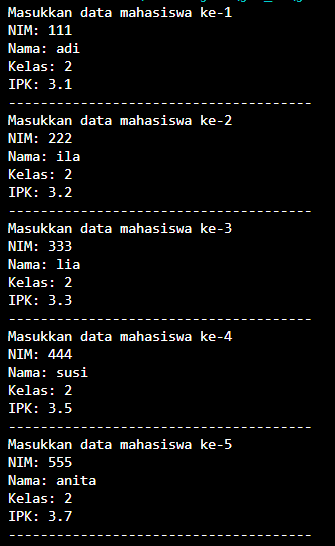
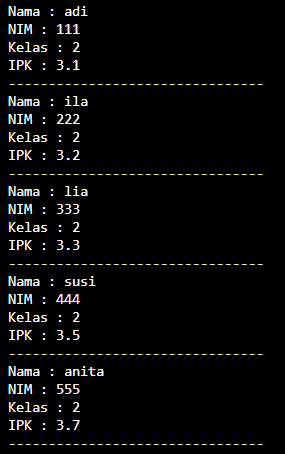
**Kode program MahasiswaBerprestasi03.java**

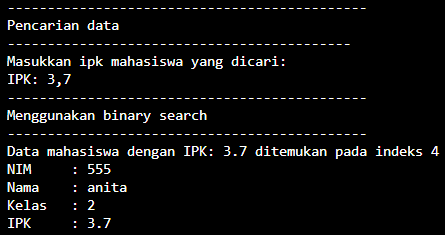
|  |
| --- |
| public class MahasiswaBerprestasi03 {      Mahasiswa03[] listMhs = new Mahasiswa03[5];      int idx;      void tambah(Mahasiswa03 m) {          if (idx < listMhs.length) {              listMhs[idx] = m;              idx++;          } else {              System.out.println("Data sudah penuh!");          }      }      void tampil() {          for (Mahasiswa03 m : listMhs) {              m.tampilInformasi();              System.out.println("--------------------------------");          }      }      void bubbleSort() {          for (int i = 0; i < listMhs.length - 1; i++) {              for (int j = 1; j < listMhs.length - i; j++) {                  if (listMhs[j].ipk > listMhs[j - 1].ipk) {                      Mahasiswa03 tmp = listMhs[j];                      listMhs[j] = listMhs[j - 1];                      listMhs[j - 1] = tmp;                  }              }          }      }      void selectionSort() {          for (int i = 0; i < listMhs.length - 1; i++) {              int idxMin = i; // ✅ diperbaiki              for (int j = i + 1; j < listMhs.length; j++) {                  if (listMhs[j].ipk < listMhs[idxMin].ipk) {                      idxMin = j;                  }              }              Mahasiswa03 tmp = listMhs[idxMin];              listMhs[idxMin] = listMhs[i];              listMhs[i] = tmp;          }      }      void insertionSort() {          for (int i = 1; i < listMhs.length; i++) {              Mahasiswa03 temp = listMhs[i];              int j = i;              while (j > 0 && listMhs[j - 1].ipk > temp.ipk) {                  listMhs[j] = listMhs[j - 1];                  j--;              }              listMhs[j] = temp;          }      }      int sequentialSerching(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");          }      }      int findBinarySearch(double cari, int left, int right) {          int mid;          if (right >= left) {              mid = (left + right) / 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;      }  } |

**Kode program MahasiswaDemo03.java**

|  |
| --- |
| import java.util.Scanner;  public class MahasiswaDemo03 {      public static void main(String[] args) {          MahasiswaBerprestasi03 list = new MahasiswaBerprestasi03();          Scanner sc  = new Scanner(System.in);          int jumMhs=5;          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: ");              String ip =sc.nextLine();              Double ipk = Double.parseDouble(ip);              System.out.println("--------------------------------------");              list.tambah(new Mahasiswa03(nim, nama, kelas, ipk));          }          list.tampil();          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();          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);      }  } |

**Hasil kode**



**PERTANYAAN**

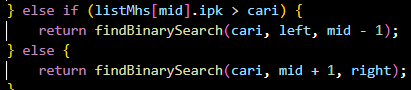
1. Tunjukkan pada kode program yang mana proses divide dijalankan!

* Program Divie terjadi ketika nilai mid dari rentang pencarian dihitung dengan rumus :



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

* Proses conquer terjadi dalam pemanggilan rekursif berdasarkan perbandingan



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

* Program masih tetap bisa berjalan tetapi tidak menemukan data yang di cari walaupun data tersebut ada

1. 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

* Mengganti if (listMhs[mid].ipk > cari) menjadi if (listMhs[mid].ipk < cari)



Menjadi



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

* Modifikasi



Kode program yang telah dimodifikasi

|  |
| --- |
| import java.util.Scanner;  public class MahasiswaDemo03 {      public static void main(String[] args) {          MahasiswaBerprestasi03 list = new MahasiswaBerprestasi03();          Scanner sc  = new Scanner(System.in);            System.out.print("Masukkan jumlah mahasiswa: ");          int jumMhs = Integer.parseInt(sc.nextLine());          for(int i=0; i<jumMhs; i++){                System.out.println("Masukkan data mahasiswa ke-" + (i + 1));              System.out.println("NIM: ");              String nim = sc.nextLine();              System.out.println("Nama: ");              String nama = sc.nextLine();              System.out.println("Kelas: ");              String kelas = sc.nextLine();              System.out.println("IPK: ");              String ip =sc.nextLine();              Double ipk = Double.parseDouble(ip);              System.out.println("--------------------------------------");              list.tambah(new Mahasiswa03(nim, nama, kelas, ipk));          }          list.tampil();          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();          System.out.println("Menggunakan sequential searching");          double posisi = list.sequentialSerching(cari);          int pss = (int)posisi;          list.tampilPosisi(cari, pss);          list.tampilDataSearch(cari, pss);      }  } |

LATIHAN PRAKTIKUM

Kode program

Kode Dosen03.java

|  |
| --- |
| public class Dosen03 {      String kode;      String nama;      boolean jenisKelamin;      int usia;      Dosen03(String kd, String name, boolean jk, int age) {          kode = kd;          nama = name;          jenisKelamin = jk;          usia = age;      }      void tampil() {          System.out.println("Kode           : " + kode);          System.out.println("Nama           : " + nama);          System.out.println("Jenis Kelamin  : " + (jenisKelamin ? "Laki-laki" : "Perempuan"));          System.out.println("Usia           : " + usia);          System.out.println("----------------------------------");      }  } |

Kode DataDosen03.java

|  |
| --- |
| public class DataDosen03 {      Dosen03[] dataDosen = new Dosen03[10];      int idx;      void tambah(Dosen03 dsn) {          if (idx < dataDosen.length) {              dataDosen[idx++] = dsn;          } else {              System.out.println("Data Dosen Sudah Penuh!");          }      }      void tampil() {          if (idx == 0) {              System.out.println("Tidak Ada Data Dosen.");              return;          }          for (int i = 0; i < idx; i++) {              dataDosen[i].tampil();          }      }      void sortingASC() {          for (int i = 0; i < idx - 1; i++) {              for (int j = 0; j < idx - 1 - i; j++) {                  if (dataDosen[j].usia > dataDosen[j + 1].usia) {                      Dosen03 temp = dataDosen[j];                      dataDosen[j] = dataDosen[j + 1];                      dataDosen[j + 1] = temp;                  }              }          }      }      void sortingDSC() {          for (int i = 0; i < idx - 1; i++) {              for (int j = 0; j < idx - 1 - i; j++) {                  if (dataDosen[j].usia < dataDosen[j + 1].usia) {                      Dosen03 temp = dataDosen[j];                      dataDosen[j] = dataDosen[j + 1];                      dataDosen[j + 1] = temp;                  }              }          }      }      void pencarianDataSequential(String nama) {          boolean ditemukan = false;          int hitung =0;          for (int i = 0; i < idx; i++) {              if (dataDosen[i].nama.equalsIgnoreCase(nama)) {                  dataDosen[i].tampil();                  ditemukan = true;                  hitung++;              }          }          if (!ditemukan) {              System.out.println("Data dengan nama '" + nama + "' tidak ditemukan.");          }      }        void pencarianDataBinary(int usiaCari) {          sortingASC();          int left = 0, right = idx - 1;          boolean ditemukan = false;            while (left <= right) {              int mid = (left + right) / 2;              if (dataDosen[mid].usia == usiaCari) {                  int awal = mid, akhir = mid;                    while (awal - 1 >= 0 && dataDosen[awal - 1].usia == usiaCari) {                      awal--;                  }                  while (akhir + 1 < idx && dataDosen[akhir + 1].usia == usiaCari) {                      akhir++;                  }                    int jumlah = akhir - awal + 1;                  System.out.println("Ditemukan " + jumlah + " data dengan usia " + usiaCari + ":");                  for (int i = awal; i <= akhir; i++) {                      dataDosen[i].tampil();                  }                    ditemukan = true;                  break;              } else if (usiaCari < dataDosen[mid].usia) {                  right = mid - 1;              } else {                  left = mid + 1;              }          }            if (!ditemukan) {              System.out.println("Data dengan usia " + usiaCari + " tidak ditemukan.");          }      }  } |

Kode DosenMain03.java

|  |
| --- |
| import java.util.Scanner;  public class DosenMain03 {      public static void main(String[] args) {          Scanner sc = new Scanner(System.in);          DataDosen03 datadsn = new DataDosen03();          int pilihan;          do {              System.out.println("\n=================== MENU ====================");              System.out.println("1. Tambah Data Dosen");              System.out.println("2. Tampilkan Data Dosen");              System.out.println("3. Cari Data Dosen (Nama - Sequential Search)");              System.out.println("4. Cari Data Dosen (Usia - Binary Search)");              System.out.println("5. Keluar");              System.out.print("Pilih Menu: ");              pilihan = sc.nextInt();              sc.nextLine();              switch (pilihan) {                  case 1:                      if (datadsn.idx >= 10) {                          System.out.println("Kapasitas data sudah penuh.");                          break;                      }                      System.out.print("Masukkan jumlah data yang ingin ditambahkan: ");                      int jumlah = sc.nextInt();                      sc.nextLine();                      for (int i = 0; i < jumlah; i++) {                          if (datadsn.idx >= 10) {                              System.out.println("Kapasitas data sudah penuh.");                              break;                          }                          System.out.println("Data Dosen ke-" + (datadsn.idx + 1));                          System.out.print("Kode Dosen         : ");                          String kode = sc.nextLine();                          System.out.print("Nama Dosen         : ");                          String nama = sc.nextLine();                          System.out.print("Jenis Kelamin (L/P): ");                          char jk = sc.next().toUpperCase().charAt(0);                          boolean jenisKelamin = jk == 'L';                          System.out.print("Usia               : ");                          int usia = sc.nextInt();                          sc.nextLine();                          Dosen03 dsn = new Dosen03(kode, nama, jenisKelamin, usia);                          datadsn.tambah(dsn);                          System.out.println("----------------------------------");                      }                      break;                  case 2:                      datadsn.tampil();                      break;                  case 3:                      System.out.print("Masukkan nama dosen yang dicari: ");                      String namaCari = sc.nextLine();                      datadsn.pencarianDataSequential(namaCari);                      break;                    case 4:                      System.out.print("Masukkan usia dosen yang dicari: ");                      int usiaCari = sc.nextInt();                      sc.nextLine();                      datadsn.pencarianDataBinary(usiaCari);                      break;                    case 5:                      System.out.println("Keluar dari program.");                      break;                  default:                      System.out.println("Pilihan tidak valid. Coba lagi.");                }          } while (pilihan != 5);          sc.close();      }  } |

**Hasil kode**

