LAPORAN HASIL PRAKTIKUM

Algoritma Dan Struktur Data

Sorting

NAMA : Handino Asa Galih r

NIM : 244107020237 KELAS

: 1-E

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLINEMA

2025

Praktikum 1

a. Sorting – Bubble Sort

Kode Program Class

```
public class Sorting12 {


    int [] data;
    int jumData;


    Sorting12 (int Data[], int jmlData) {
        jumData = jmlData;
        data = new int [jmlData];
        for (int i = 0; i < jmlData; i++) {
            data[i] = Data[i];
        }
    }
    void bubbleSort() {
        int temp = 0;
        for (int i = 0; i < jumData; i++) {
            for (int j = 1; j < jumData - i; j++) {
                if (data[j-1] > data[j]) {
                    temp = data[j];
                    data[j] = data[j - 1];
                    data[j - 1] = temp;
                }
            }
        }
    }
    void tampil() {
        for (int i = 0; i < jumData; i++) {
            System.out.print(data[i] + " ");
```

```
        }
        System.out.println();
    }
}
```

Kode Program Main

```
public class SortingMain12 {     public
static void main(String[] args) {
int a[] = {20, 10, 2, 7, 12};



        Sorting12 dataurut1 = new Sorting12(a, a.length);
        System.out.println("Data awal 1");
dataurut1.tampil();
dataurut1.bubbleSort();
        System.out.println("Data sudah diurutkan dengan BUBBLE SORT
(ASC)");
dataurut1.tampil();
    }
}
```

Output

```
Data awal 1
20 10 2 7 12
Data sudah diurutkan dengan BUBBLE SORT (ASC)
2 7 10 12 20
```

b. Sorting- Selection Sort

Kode Program Class (Penambahan)

```
void selectionSort() {
        for (int i = 0; i < jumData - 1; i++) {
            int min = i;
            for (int j = i + 1; j < jumData; j++) {
                if (data[j] < data[min]) {
                    min = j;
                }
            }
            int temp = data[i];
            data[i] = data[min];
            data[min] = temp;
        }
    }
```

Kode Program Main

```
public class SortingMain12 {
    public static void main(String[] args) {
        int a[] = {20, 10, 2, 7, 12};
        int b[] = {30, 20, 2, 8, 14};

        Sorting12 dataurut1 = new Sorting12(a, a.length);
        Sorting12 dataurut2 = new Sorting12(b, b.length);

        System.out.println("Data awal 1");
        dataurut1.tampil();
        dataurut1.bubbleSort();
        System.out.println("Data sudah diurutkan dengan BUBBLE SORT
(ASC)");
        dataurut1.tampil();

        System.out.println("Data awal 2");
```

```
      Dataurut2.tampil();
        Dataurut2.insertionSort();
        System.out.println("Data sudah diurutkan dengan INSERTION SORT
(ASC)");
        Dataurut2.tampil();
    }
}
```

Output

```
Data awal 2
30 20 2 8 14
Data sudah diurutkan dengan SELECTION SORT (ASC)
2 8 14 20 30
```

c. Sorting – Insertion sort

Kode Program Class (Penambahan)

```
  void insertionSort() {
        for (int i = 1; i <= data.length-1; i++) {
            int temp = data[i];
            int j = i - 1;
            while (j>=0 && data [j]>temp) {
                data [j+1] = data [j];
                j--;
            }
            data [j+1] = temp;
        }
    }
```

Kode Program Main

```
  public class SortingMain12 {
    public static void main(String[] args) {
        int a[] = {20, 10, 2, 7, 12};
        int b[] = {30, 20, 2, 8, 14};
        int c[] = {40, 10, 4, 9, 3};

        Sorting12 dataurut1 = new Sorting12(a, a.length);
        Sorting12 dataurut2 = new Sorting12(b, b.length);
        Sorting12 dataurut3 = new Sorting12(c, c.length);

        System.out.println("Data awal 1");
        dataurut1.tampil();
        dataurut1.bubbleSort();
        System.out.println("Data sudah diurutkan dengan BUBBLE SORT
(ASC)");
        dataurut1.tampil();

        System.out.println("Data awal 2");
        dataurut2.tampil();
        dataurut2.selectionSort();
        System.out.println("Data sudah diurutkan dengan SELECTION SORT
(ASC)");
        dataurut2.tampil();

        System.out.println("Data awal 3");
        dataurut3.tampil();
        dataurut3.insertionSort();
        System.out.println("Data sudah diurutkan dengan INSERTION SORT
(ASC)");
        dataurut3.tampil();
    }
}
```

Output

```
Data awal 3
40 10 4 9 3
Data sudah diurutkan dengan INSERTION SORT (ASC)
3 4 9 10 40
```

Pertanyaan

1. **Jelaskan fungsi kode program berikut**

```
if (data[j-1]>data[j]){
    temp=data[j];
    data[j]=data[j-1];
    data[j-1]=temp;
}
```

Kode ini digunakan untuk menukar (switch) posisi dua elemen yang bersebelahan dalam array jika elemen sebelumnya (data[j-1]) lebih besar dari elemen (data[j]). Proses pertukaran :

1. int temp = data[j] => Nilai data[j] disimpan sementara di variable temp.
2. data[j] = data [j-1] => Nilai data[j-1] dipindahkan ke posisi data[j].
3. data[j-1] = temp => Nilai temp (yang merupakan nilai data[j] semula) dipindahkan ke posisi data [j-1].

2. Tunjukkan kode program yang merupakan algoritma pencarian nilai minimum pada selction sort!

```
int min = i;
for (int j = i + 1; j < jumData; j++) {
    if (data[j] < data[min]) {
        min = j;
    }
}
```

Kode tersebut merupakan kode program untuk mencari nilai minimum pada selection sort. Program ini melakukakn Inisialisasi min dengan indeks elemen pertama dari bagian array yang belum diurutkan. Lalu, membandingkan elemen-elemen yang lebih kecil. Setelah loop dalam selesai, min akan berisi indeks dari elemen minimum dalam bagian array yang belum diurutkan.

3. Pada insertion sort, jelaskan maksud dari kondisi pada perulangan

```
while (j>=0 && data[j]>temp)
```

Kondisi ini digunakan untuk menyisipkan elemen temp (elemen yang diurutkan) ke dalam bagian array yang sudah terurut. Loop while ini terus berjalan selama kita belum mencapai awal array (j>=0) dan selama elemen-elemen di sebelah kiri temp masih lebih beasr dari tep (data[j] > temp). Tujuannya adalah untuk menemukan posisi yang lebih tepat untuk menyisipkan temp ke dalam bagian array yang sudah terurut dengan benar.

4. Pada Insertion sort, apakah tujuan dari perintah

```
data[j+1]= data[j];
```

Tujuan utama dari perintah data [j+1] = data[j]; pada Insertion Sort adalah untuk mengisi nilai pada array data indeks ke (j+1) dengan nilai array indeks ke-j. Oleh karena itu, elemen data[j] akan geser satu posisi ke kanan (ke indeks j+1) dalam array.

Praktikum 2

Kode Program Class Mahasiswa12

```
public class Mahasiswa12 {
    String nim;
    String nama;
    String kelas;
    double ipk;

    Mahasiswa12 () {
    }

    Mahasiswa12 (String nm, String name, String kls, double
ip) {
        nim = nm;
        nama = name;
        ipk = ip;
        kelas = kls;
    }

    void tampilInformasi() {
        System.out.println("Nama : " + nama);
        System.out.println("NIM : " + nim);
        System.out.println("Kelas : " + kelas);
        System.out.println("IPK : " + ipk);
    }
}
```

Kode Program Class MahasiswaBerprestasi12

```
public class MahasiswaBerprestasi12 {
    Mahasiswa12 [] listMhs = new Mahasiswa12[5];
    int idx;
```

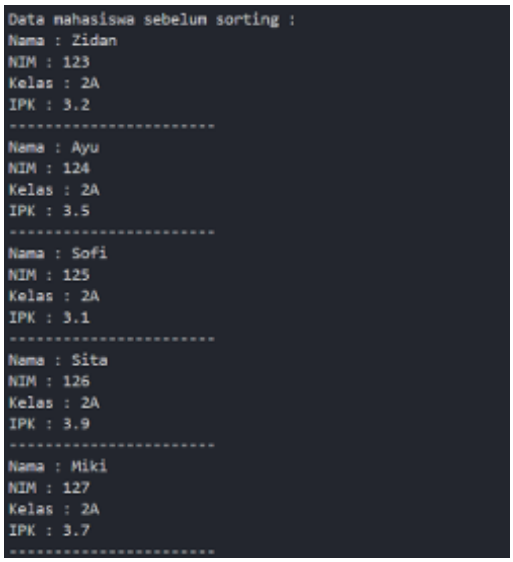
```
 void tambah (Mahasiswa12 m) {
        if (idx < listMhs.length) {
            listMhs [idx] = m;
            idx++;
        } else {
            System.out.println ("Data sudah penuh");
        }
    }

    void tampil() {
        for (Mahasiswa12 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) {
                    Mahasiswa12 tmp = listMhs[j];
                    listMhs[j] = listMhs[j-1];
                    listMhs[j-1] = tmp;
                }
            }
        }
    }
```

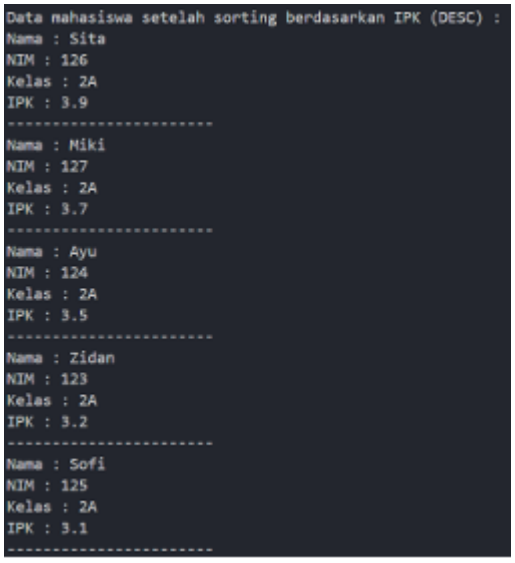
Kode Program Main MahasiswaDemo12

```
import java.util.Scanner;
public class MahasiswaDemo12 {
    public static void main(String[] args) {
```

```
        MahasiswaBerprestasi12 list = new
MahasiswaBerprestasi12();

        Mahasiswa12 m1 = new Mahasiswa12("123", "Zidan",
"2A", 3,2);

        Mahasiswa12 m2 = new Mahasiswa12("124", "Ayu", "2A",
3,5);

        Mahasiswa12 m3 = new Mahasiswa12("125", "Sofi", "2A",
3,1);

        Mahasiswa12 m4 = new Mahasiswa12("126", "Sita", "2A",
3,9);

        Mahasiswa12 m5 = new Mahasiswa12("127", "Miki", "2A",
3,7);

            list.tambah(m1);

            list.tambah(m2);

            list.tambah(m3);

            list.tambah(m4);

            list.tambah(m5);


        System.out.println("Data Mahasiswa sebelum sorting :
");

        list.tampil();


        System.out.println("Data Mahasiswa setelah sorting
berdasarkan IPK (DESC) : ");

        list.bubbleSort();

        list.tampil();

    }

}
```

Output





Pertanyaan

1. Perhatikan perulangan di dalam bubbleSort() di bawah ini :

```
for (int i=0; i<listMhs.length-1; i++){
    for (int j=1; j<listMhs.length-i; j++){
```

   a. Mengapa syarat dari perulangan I adalah i<ListMhs.length-1 ?

      Pertama, dalam setiap iterasi loop luar, elemen terbesar (atau terkecil, tergantung pada urutan pengurutan) akan berpindah ke posisi akhirnya di bagian akhir array. Kedua setelah listMhs.length -1 iterasi, elemen terakhir dalam array pasti sudah berada di posisi yang benar, sehingga iterasi terakhir menjadi tidak perlu.
      Ketiga, Batasan ini mencegah terjadinya ArrayIndexOutOfBoundsException. Jika kita menjalankan iterasi hingga i< listMhs.length, maka pada iterasi terakhir loop dalam, variable j bisa mencapai listMhs.length, yangakan menyebabkan kesalahan kerena indeks array dimulai dari 0 dan berakhir di listMhs.length – 1.

   b. Mengapa syarat dari perulangan j adalah j<listMhs.length-i ?

      Pertama, pada setiap iterasi loop luar (loop i), elemen terakhir dari array sudah berada di posisi yang benar, yang berarti mereka sudah teurut. Oleh karena itu, tidak perlu lagi membandingkan elemen elemen di bagian array yang sudah terurut.
      Kedua, dengan mengurangi jumlah iterasi loop dalam sebanyak i, hal ini menghindari perbandingan yang tidak perlu sehingga algoritma menjadi lebih efisien.

   c. Jika banyak data di dalam listMhs adalah 50, maka berapakali perulangan i akan berlangsung? Dan ada berapa Tahap bubbleSort yang ditempuh?

      Jika listMhs.length adalah 50, maka perulangan I akan berlangsung sebanyak listMhs.length -1 kali, yaitu 49 kali (I akan berjalan dari 0 hingga 48). Jumlah tahap bubble

Sort sama dengan jumlah perulangan i, yaitu 49 tahap, Setiap tahap akan menempatkan satu elemen pada posisi yang benar di bagian akhir array.

2. Modifikasi program diatas Dimana data mahasiswa bersifat dinamis (input dari keyboard) yang terdiri dari nim, nama, kelas, dan ipk!

Kode program yang dimodifikasi pada MahasiswaDemo12.java :

```
import java.util.Scanner;

public class MahasiswaDemo12 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        MahasiswaBerprestasi12 list = new
MahasiswaBerprestasi12();


        for (int i = 0; i < 5; i++) {

            System.out.println("Masukkan Data Mahasiswa ke-" +
(i + 1));

            System.out.print("NIM     : ");

            String nim = sc.next();

            System.out.print("Nama    : ");

            String nama = sc.next();

            System.out.print("Kelas   : ");

            String kelas = sc.next();

            System.out.print("IPK     : ");

            double ipk = sc.nextDouble();


            Mahasiswa12 m = new Mahasiswa12(nim, nama, kelas,
ipk);

            list.tambah(m);

            System.out.println();

        }


        System.out.println("Data Mahasiswa (belum diurutkan):
");

        list.tampil();

        System.out.println();
```

```
        System.out.println("Data Mahasiswa Berprestasi
(Diurutkan berdasarkan IPK) : ");

        list.tampil();

    }

}
```

Output

```
Masukkan Data Mahasiswa ke-1
NIM     : 123
Nama    : Zidan
Kelas   : 2A
IPK     : 3,2

Masukkan Data Mahasiswa ke-2
NIM     : 124
Nama    : Ayu
Kelas   : 2A
IPK     : 3,5

Masukkan Data Mahasiswa ke-3
NIM     : 125
Nama    : Sofi
Kelas   : 2A
IPK     : 3,1

Masukkan Data Mahasiswa ke-4
NIM     : 126
Nama    : Sita
Kelas   : 2A
IPK     : 3,9

Masukkan Data Mahasiswa ke-5
NIM     : 127
Nama    : Miki
Kelas   : 2A
IPK     : 3,7
```

```
Data Mahasiswa (belum diurutkan):
Nama : Zidan
NIM : 123
Kelas : 2A
IPK : 3.2
-----------------------
Nama : Ayu
NIM : 124
Kelas : 2A
IPK : 3.5
-----------------------
Nama : Sofi
NIM : 125
Kelas : 2A
IPK : 3.1
-----------------------
Nama : Sita
NIM : 126
Kelas : 2A
IPK : 3.9
-----------------------
Nama : Miki
NIM : 127
Kelas : 2A
IPK : 3.7
-----------------------
```

```
Data Mahasiswa Berprestasi (diurutkan berdasarkan IPK):
Nama : Sita
NIM : 126
Kelas : 2A
IPK : 3.9
----------------------
Nama : Miki
NIM : 127
Kelas : 2A
IPK : 3.7
----------------------
Nama : Ayu
NIM : 124
Kelas : 2A
IPK : 3.5
----------------------
Nama : Zidan
NIM : 123
Kelas : 2A
IPK : 3.2
----------------------
Nama : Sofi
NIM : 125
Kelas : 2A
IPK : 3.1
----------------------
```

Mengurutkan Data Mahasiswa Berdasarkan IPK (Selection Sort)

Kode Program Class MahasiswaBerprestasi12 (penambahan)

```
    void selectionSort() {
        for (int i = 0; i < listMhs.length-1; i++) {
            int idxMin = i;
            for (int j = i+1; j < listMhs.length; j++) {
                if (listMhs[j].ipk<listMhs[idxMin].ipk) {
                    idxMin = j;
                }
            }
            Mahasiswa12 tmp = listMhs[idxMin];
            listMhs [idxMin] = listMhs[i];
            listMhs [i] = tmp;
        }
    }
```

Kode Program Class MahasiswaDemo12 (penambahan)

```
      System.out.println("Data yang sudah terurut menggunakan
SELETION SORT (ASC) : "):

      list.selectionSort();

      list.tampil()
```

Output

```
Masukkan Data Mahasiswa ke-1
NIM     : 123
Nama    : Ali
Kelas   : 2B
IPK     : 3,9

Masukkan Data Mahasiswa ke-2
NIM     : 124
Nama    : Lila
Kelas   : 2B
IPK     : 3,1

Masukkan Data Mahasiswa ke-3
NIM     : 125
Nama    : Agus
Kelas   : 2B
IPK     : 3,6

Masukkan Data Mahasiswa ke-4
NIM     : 126
Nama    : Tika
Kelas   : 2B
IPK     : 3,3

Masukkan Data Mahasiswa ke-5
NIM     : 127
Nama    : Udin
Kelas   : 2B
IPK     : 3,2
```

```
Data yang sudah terurut menggunakan SELECTION SORT (ASC) :
Nama : Lila
NIM : 124
Kelas : 2B
IPK : 3.1
----------------------
Nama : Udin
NIM : 127
Kelas : 2B
IPK : 3.2
----------------------
Nama : Tika
NIM : 126
Kelas : 2B
IPK : 3.3
----------------------
Nama : Agus
NIM : 125
Kelas : 2B
IPK : 3.6
----------------------
Nama : Ali
NIM : 123
Kelas : 2B
IPK : 3.9
----------------------
```

Pertanyaan

Di dalam method selectionSort, terdapat baris program seperti dibawah ini :

```
int idxMin=i;
for (int j=i+1; j<listMhs.length; j++){
    if (listMhs[j].ipk<listMhs[idxMin].ipk){
        idxMin=j;
    }
}
```

Untuk Apakah proses tersebut, Jelaskan!

Digunakan untuk mengurutkan data berdasarkan IPK. Prosesnya dimulai dengan mengasumsikan elemen pertama sebagai elemen dengan IPK terkecil. Kemudian, kode tersebut membandingkan IPK elemen tersebut dengan elemen tersebut disimpan. Setelah seluruh elemen dibandingkan, indeks elemen dengan IPK terkecil akan digunakan untuk menukar posisi elemen tersebut dengan elemen pertama. Proses ini diulang untuk setiap elemen, secara bertahap membangun daftar yang terurut berdasarkan IPK.

Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan InsertionSort

Kode Program Class MahasiswaBerprestasi12(penambahan)

```
    void insertionSort() {
        for (int i = 1; i < listMhs.length; i++) {
            Mahasiswa12 temp = listMhs[i];
            int j = i;
            while (j>0 && listMhs[j-1].ipk<temp.ipk) {
                listMhs[j] = listMhs[j-1];
                j--;
            }
            listMhs[j] = temp;
        }
    }
}
```

Kode Program Class MahasiswaDemo12 (penambahan)

```
      System.out.println("Data yang sudah terurut menggunakan
INSERTION SORT (ASC) : ");

      list.insertionSort();

      list.tampil();
```

Output

```
Masukkan Data Mahasiswa ke-1
NIM     : 111
Nama    : Ayu
Kelas   : 2C
IPK     : 3,7

Masukkan Data Mahasiswa ke-2
NIM     : 222
Nama    : Dika
Kelas   : 2C
IPK     : 3

Masukkan Data Mahasiswa ke-3
NIM     : 333
Nama    : Ila
Kelas   : 2C
IPK     : 3,8

Masukkan Data Mahasiswa ke-4
NIM     : 444
Nama    : Suci
Kelas   : 2C
IPK     : 3,1

Masukkan Data Mahasiswa ke-5
NIM     : 555
Nama    : Yayuk
Kelas   : 2C
IPK     : 3,4
```

```
Data yang sudah terurut menggunakan INSERTION SORT (ASC) :
Nama : Dika
NIM : 222
Kelas : 2C
IPK : 3.0
----------------------
Nama : Suci
NIM : 444
Kelas : 2C
IPK : 3.1
----------------------
Nama : Yayuk
NIM : 555
Kelas : 2C
IPK : 3.4
----------------------
Nama : Ayu
NIM : 111
Kelas : 2C
IPK : 3.7
----------------------
Nama : Ila
NIM : 333
Kelas : 2C
IPK : 3.8
----------------------
```

Pertanyaan

Ubahlah fungsi pada InsertionSort sehingga fungsi ini dapat melaksanakan proses sorting dengan cara descending.

Kode Program Class MahasiswaBerprestasi12 (penambahan)

```
    void insertionSort() {
        for (int i = 1; i < listMhs.length; i++) {
            Mahasiswa12 temp = listMhs[i];
            int j = i;
            while (j>0 && listMhs[j-1].ipk<temp.ipk) {
                listMhs[j] = listMhs[j-1];
                j--;
            }
            listMhs[j] = temp;
        }
    }
```

Output

```
Data yang sudah terurut menggunakan INSERTION SORT (ASC) :
Nama : Ila
NIM : 333
Kelas : 2C
IPK : 3.8
----------------------
Nama : Ayu
NIM : 111
Kelas : 2C
IPK : 3.7
----------------------
Nama : Yayuk
NIM : 555
Kelas : 2C
IPK : 3.4
----------------------
Nama : Suci
NIM : 444
Kelas : 2C
IPK : 3.1
----------------------
Nama : Dika
NIM : 222
Kelas : 2C
IPK : 3.0
----------------------
```

Latihan Praktikum

Kode Program Dosen12.javs

```
public class Dosen12 {
    String kode;
    String nama;
    boolean jenisKelamin;
    int usia;

    Dosen12() {
    }

    Dosen12(String kd, String name, boolean jk, int age) {
        kode = kd;
        nama = name;
        jenisKelamin = jk;
        usia = age;
    }

    void tampilData() {
        System.out.println("Kode          : " + kode);
        System.out.println("Nama          : " + nama);
        if (jenisKelamin == true) {
            System.out.println("Jenis Kelamin : Wanita");
        } else {
            System.out.println("Jenis Kelamin : Pria");
        }
        System.out.println("Usia          : " + usia);
    }
}
```

Kode Program DataDosen12.java

```
public class DataDosen12 {
    Dosen12[] dataDosen;
    int idx;

    public DataDosen12(int jumlah) {
        dataDosen = new Dosen12[jumlah];
    }

    void tambah(Dosen12 dsn) {
        if (idx < dataDosen.length) {
            dataDosen[idx] = dsn;
            idx++;
        } else {
            System.out.println("data sudah penuh");
        }
    }

    void tampil() {
        for (Dosen12 dsn:dataDosen) {
            dsn.tampilData();
            System.out.println("------------------------------");
        }
    }

    void SortingASC() {
        for (int i = 0; i < dataDosen.length-1; i++) {
            for (int j = 1; j < dataDosen.length-i; j++) {
                if (dataDosen[j].usia < dataDosen[j-1].usia) {
                    Dosen12 tmp = dataDosen[j];
                    dataDosen[j] = dataDosen[j-1];
                    dataDosen[j-1] = tmp;
                }
```

```
    void SortingDSC() {
        for (int i = 0; i < (dataDosen.length-1); i++) {
            int idxMin = i;
            for (int j = (i+1); j < dataDosen.length; j++) {
                if (dataDosen[j].usia > dataDosen[idxMin].usia) {
                    idxMin = j;
                }
            }
            Dosen12 tmp = dataDosen[idxMin];
            dataDosen[idxMin] = dataDosen[i];
            dataDosen[i] = tmp;
        }
    }

    void insertionSort() {
        for (int i = 1; i < dataDosen.length; i++) {
            Dosen12 temp = dataDosen[i];
            int j = i;
            while (j > 0 && dataDosen[j-1].usia < temp.usia) {
                dataDosen[j] = dataDosen[j-1];
                j--;
            }
            dataDosen[j] = temp;
        }
    }
```

Kode Program DosenDemo12.java

```
import java.util.Scanner;
public class DosenDemo12{
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        DataDosen12 dosen = new DataDosen12();
            for (int i = 0; i < 10; i++) {
                System.out.println("Masukkan data dosen ke-" + (i+1));
                System.out.print("Masukkan Kode          : ");
                String kode = input.next();
                System.out.print("Masukkan Nama          : ");
                String nama = input.next();
                System.out.print("Jenis Kelamin (Pria/Wanita) : ");
                String jenisKelamin = sc.nextLine();
                boolean jk = false;
                if (jenisKelamin.equalsIgnoreCase("wanita")) {
                    jk = true;
                }
                System.out.print("Masukkan Usia          : ");
                int usia = sc.nextInt();
                sc.nextLine();
                Dosen12 dsn = new Dosen12(kode, nama, jk, usia);
                dosen.tambah(dsn);
                System.out.println();
            }


            System.out.println("Data dosen sebelum sorting ");
            dosen.tampil();


            System.out.println();
            System.out.println("Data dosen setelah sorting menggunakan
BUBBLE SORT (ASC) : ");
```

```
            System.out.println();

            System.out.println("Data dosen setelah sorting menggunakan
SELECTION SORT (DSC) : ");

            Dosen.sortingDSC();

            dosen.tampil();



            System.out.println();

            System.out.println("Data dosen setelah sorting mnenggunakan
INSERTION SORT (DSC) : ");

            Dosen.isertionSort();

            Dosen.tampil();

    }

}
```

Output

```
Masukkan jumlah dosen : 10
Masukkan jumlah data dosen ke-1
Masukkan Kode          : 1
Masukkan Nama          : Nikma
Masukkan Jenis Kelamin : wanita
Masukkan Usia          : 21

Masukkan jumlah data dosen ke-2
Masukkan Kode          : 2
Masukkan Nama          : Dino
Masukkan Jenis Kelamin : Pria
Masukkan Usia          : 24

Masukkan jumlah data dosen ke-3
Masukkan Kode          : 3
Masukkan Nama          : Rani
Masukkan Jenis Kelamin : wanita
Masukkan Usia          : 22

Masukkan jumlah data dosen ke-4
Masukkan Kode          : 4
Masukkan Nama          : Ghani
Masukkan Jenis Kelamin : pria
Masukkan Usia          : 25

Masukkan jumlah data dosen ke-5
Masukkan Kode          : 5
Masukkan Nama          : Sita
Masukkan Jenis Kelamin : wanita
Masukkan Usia          : 22
```

```
Masukkan jumlah data dosen ke-6
Masukkan Kode          : 6
Masukkan Nama          : Budi
Masukkan Jenis Kelamin : pria
Masukkan Usia          : 28

Masukkan jumlah data dosen ke-7
Masukkan Kode          : 7
Masukkan Nama          : cici
Masukkan Jenis Kelamin : wanita
Masukkan Usia          : 23

Masukkan jumlah data dosen ke-8
Masukkan Kode          : 8
Masukkan Nama          : Dani
Masukkan Jenis Kelamin : pria
Masukkan Usia          : 21

Masukkan jumlah data dosen ke-9
Masukkan Kode          : 9
Masukkan Nama          : Dinda
Masukkan Jenis Kelamin : wanita
Masukkan Usia          : 23

Masukkan jumlah data dosen ke-10
Masukkan Kode          : 10
Masukkan Nama          : Dana
Masukkan Jenis Kelamin : pria
Masukkan Usia          : 27
```

```
----------------------------------------
Data dosen sebelum sorting
----------------------------------------
Kode          : 1
Nama          : Nikma
Jenis Kelamin : Wanita
Usia          : 21
-----------------------------
Kode          : 2
Nama          : Dino
Jenis Kelamin : Pria
Usia          : 24
-----------------------------
Kode          : 3
Nama          : Rani
Jenis Kelamin : Wanita
Usia          : 22
-----------------------------
Kode          : 4
Nama          : Ghani
Jenis Kelamin : Pria
Usia          : 25
-----------------------------
Kode          : 5
Nama          : Sita
Jenis Kelamin : Wanita
Usia          : 22
-----------------------------
Kode          : 6
Nama          : Budi
Jenis Kelamin : Pria
Usia          : 28
-----------------------------
Kode          : 7
Nama          : cici
Jenis Kelamin : Wanita
-----------------------------
Kode          : 8
Nama          : Dani
Jenis Kelamin : Pria
Usia          : 21
-----------------------------
Kode          : 9
Nama          : Dinda
Jenis Kelamin : Wanita
Usia          : 23
-----------------------------
Kode          : 10
Nama          : Dana
Jenis Kelamin : Pria
Usia          : 27
-----------------------------
----------------------------------------
Searching menggunakan Sequential Search
----------------------------------------
Masukkan nama dosen yang dicari : Dino
Data dosen dengan nama : Dino ditemukan pada indeks 1
Dengan detail data dosen :
Kode          : 2
Nama          : Dino
Jenis Kelamin : Pria
Usia          : 24

================================================
PERINGATAN : Data ditemukan sebanyak 1
================================================
----------------------------------------
Searching menggunakan Binary Search
----------------------------------------
Masukkan usia yang dicari : 22
================================================
PERINGATAN : Data ditemukan sebanyak 2
```

```
================================================
Data dosen dengan usia : 22 ditemukan pada indeks 2
Dengan detail data dosen :
Kode          : 3
Nama          : Rani
Jenis Kelamin : Wanita
Usia          : 22

Data dosen dengan usia : 22 ditemukan pada indeks 3
Dengan detail data dosen :
Kode          : 5
Nama          : Sita
Jenis Kelamin : Wanita
Usia          : 22

PERINGATAN: Terdapat lebih dari satu data dosen dengan
usia tersebut.
PS C:\TUGAS ALSD\Praktikum ASD\jobsheet6>
```