

LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA
JOBSHEET 6



ATHAULLA HAFIZH

244107020030

TI 1 E

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

2024

Praktikum 1

a. Sorting – Bubble Sort

Kode Program Class

```
package Jobsheet6;

public class Sorting05 {

    int [] data;
    int jumData;

    Sorting05 (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

```
package Jobsheet6;  
  
public class SortingMain05 {  
    public static void main(String[] args) {  
        int a[] = {20, 10, 2, 7, 12};  
  
        Sorting05 dataurut1 = new Sorting05(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

```
package Jobsheet6;

public class SortingMain05 {
    public static void main(String[] args) {
        int a[] = {20, 10, 2, 7, 12};
        int b[] = {30, 20, 2, 8, 14};

        Sorting05 dataaurut1 = new Sorting05(a, a.length);
        Sorting05 dataaurut2 = new Sorting05(b, b.length);

        System.out.println("Data awal 1");
        dataaurut1.tampil();
        dataaurut1.bubbleSort();
        System.out.println("Data sudah diurutkan dengan BUBBLE SORT (ASC)");
        dataaurut1.tampil();

        System.out.println("Data awal 2");
        dataaurut2.tampil();
    }
}
```

```

        dataurut2.selectionSort();

        System.out.println("Data sudah diurutkan dengan SELECTION
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

```
package Jobsheet6;

public class SortingMain05 {

    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};

        Sorting05 dataaurut1 = new Sorting05(a, a.length);
        Sorting05 dataaurut2 = new Sorting05(b, b.length);
        Sorting05 dataaurut3 = new Sorting05(c, c.length);

        System.out.println("Data awal 1");
        dataaurut1.tampil();
        dataaurut1.bubbleSort();
        System.out.println("Data sudah diurutkan dengan BUBBLE SORT (ASC)");
        dataaurut1.tampil();

        System.out.println("Data awal 2");
        dataaurut2.tampil();
        dataaurut2.selectionSort();
        System.out.println("Data sudah diurutkan dengan SELECTION SORT (ASC)");
        dataaurut2.tampil();

        System.out.println("Data awal 3");
        dataaurut3.tampil();
        dataaurut3.insertionSort();
        System.out.println("Data sudah diurutkan dengan INSERTION SORT (ASC)");
        dataaurut3.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 variabel 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 selection 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 melakukan Inisialisasi min dengan indeks elemen pertama dari bagian array yang belum diurutkan. Lalu, membandingkan elemen-elemen berikutnya dengan data[min]. Kemudian, memperbarui min jika ditemukan 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 sedang diurutkan) ke dalam bagian array yang sudah terurut. Loop while ini terus berjalan selama kita belum mencapai awal array ($j \geq 0$) dan selama elemen-elemen di sebelah kiri temp masih lebih besar dari temp ($\text{data}[j] > \text{temp}$). Tujuannya adalah untuk menemukan posisi yang 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 $\text{data}[j+1] = \text{data}[j];$ pada Insertion Sort adalah untuk mengisi nilai pada array data index ke $(j+1)$ dengan nilai array index ke- j . Oleh karena itu, elemen $\text{data}[j]$ akan geser satu posisi ke kanan (ke indeks $j+1$) dalam array.

Praktikum 2

Kode Program Class Mahasiswa05

```
package Jobsheet6;

public class Mahasiswa05 {

    String nim;
    String nama;
    String kelas;
    double ipk;

    Mahasiswa05 () {

    }

    Mahasiswa05 (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 MahasiswaBerprestasi05

```
package Jobsheet6;

public class MahasiswaBerprestasi05 {

    Mahasiswa05 [] listMhs = new Mahasiswa05[5];
    int idx;
```

```

void tambah (Mahasiswa05 m) {
    if (idx < listMhs.length) {
        listMhs [idx] = m;
        idx++;
    } else {
        System.out.println ("Data sudah penuh");
    }
}

void tampil() {
    for (Mahasiswa05 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) {
                Mahasiswa05 tmp = listMhs[j];
                listMhs[j] = listMhs[j-1];
                listMhs[j-1] = tmp;
            }
        }
    }
}
}

```

Kode Program Main MahasiswaDemo05

```

package Jobsheet6;

public class MahasiswaDemo05 {
    public static void main(String[] args) {

```

```

MahasiswaBerprestasi05 list = new MahasiswaBerprestasi05();
Mahasiswa05 m1 = new Mahasiswa05("123", "Zidan", "2A", 3.2);
Mahasiswa05 m2 = new Mahasiswa05("124", "Ayu", "2A", 3.5);
Mahasiswa05 m3 = new Mahasiswa05("125", "Sofi", "2A", 3.1);
Mahasiswa05 m4 = new Mahasiswa05("126", "Sita", "2A", 3.9);
Mahasiswa05 m5 = new Mahasiswa05("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

```

Data mahasiswa sebelum sorting :
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 setelah sorting berdasarkan IPK (DESC) :
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
-----

```

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 $i < \text{listMhs.length} - 1$?

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 $\text{listMhs.length} - 1$ iterasi, elemen terakhir dalam array pasti sudah berada di posisi yang benar, sehingga iterasi terakhir menjadi tidak perlu. Ketiga, batasan ini juga mencegah terjadinya `ArrayIndexOutOfBoundsException`. Jika kita menjalankan iterasi hingga $i < \text{listMhs.length}$, maka pada iterasi terakhir loop dalam, variabel j bisa mencapai listMhs.length , yang akan menyebabkan kesalahan karena indeks array dimulai dari 0 dan berakhir di $\text{listMhs.length} - 1$.

- b. Mengapa syarat dari perulangan j adalah $j < \text{listMhs.length} - i$?

Pertama, pada setiap iterasi loop luar (loop i), i elemen terakhir dari array sudah berada di posisi yang benar, yang berarti mereka sudah terurut. 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 berapakah perulangan i akan berlangsung? Dan ada berapa Tahap bubble sort yang ditempuh?

Jika listMhs.length adalah 50, maka perulangan i akan berlangsung sebanyak $\text{listMhs.length} - 1$ 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 MahasiswaDemo05.java :

```
package Jobsheet6;  
  
import java.util.Scanner;  
  
public class MahasiswaDemo05 {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        MahasiswaBerprestasi05 list = new MahasiswaBerprestasi05();
```

```
        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();  
  
            Mahasiswa05 m = new Mahasiswa05(nim, nama, kelas, ipk);  
            list.tambah(m);  
            System.out.println();  
        }  
  
        System.out.println("Data Mahasiswa (belum diurutkan): ");  
        list.tampil();  
        System.out.println();  
  
        list.bubbleSort();  
        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 MahasiswaBerprestasi05 (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;  
            }  
        }  
        Mahasiswa05 tmp = listMhs[idxMin];  
        listMhs [idxMin] = listMhs[i];  
        listMhs [i] = tmp;  
    }  
}
```

Kode Program Class MahasiswaDemo05 (penambahan)

```
System.out.println("Data yang sudah terurut menggunakan  
SELECTION 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 selection sort, terdapat baris program seperti di bawah 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 elemen setelahnya. Jika ditemukan elemen dengan IPK yang lebih kecil, indeks 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 Insertion Sort

Kode Program Class MahasiswaBerprestasi05 (penambahan)

```
void insertionSort() {
    for (int i = 1; i < listMhs.length; i++) {
        Mahasiswa05 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 MahasiswaDemo05 (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 MahasiswaBerprestasi05 (penambahan)

```
void insertionSort() {  
    for (int i = 1; i < listMhs.length; i++) {  
        Mahasiswa05 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 Dosen05.java

```
package Jobsheet6;

public class Dosen05 {

    String kode;

    String nama;

    boolean jenisKelamin;

    int usia;

    Dosen05() {

    }

    Dosen05 (String kd, String name, boolean jk, int age) {

        kode = kd;

        nama = name;

        jenisKelamin = jk;

        usia = age;

    }

    void tampilData() {

        System.out.println("Kode Dosen : " + kode);

        System.out.println("Nama Dosen : " + nama);

        if (jenisKelamin == true) {

            System.out.println("Jenis Kelamin : Wanita");

        } else {

            System.out.println("Jenis Kelamin : Pria");

        }

        System.out.println("Usia Dosen : " + usia);

    }

}
```

Kode Program DataDosen05.java

```
package Jobsheet6;

public class DataDosen05 {
    Dosen05[] dataDosen05 = new Dosen05[10];
    int idx;

    void tambah (Dosen05 dsn) {
        if (idx < dataDosen05.length) {
            dataDosen05 [idx] = dsn;
            idx++;
        } else {
            System.out.println ("Data sudah penuh");
        }
    }

    void tampil () {
        for (Dosen05 dsn:dataDosen05) {
            dsn.tampilData();
            System.out.println("-----");
        }
    }

    void sortingASC () {
        for (int i = 0; i < dataDosen05.length-1; i++) {
            for (int j = 1; j < dataDosen05.length-i; j++) {
                if (dataDosen05 [j].usia < dataDosen05 [j-1].usia) {
                    Dosen05 temp = dataDosen05 [j];
                    dataDosen05 [j] = dataDosen05 [j-1];
                    dataDosen05 [j-1] = temp;
                }
            }
        }
    }
}
```

```

void sortingDSC () {
    for (int i = 0; i < dataDosen05.length-1; i++) {
        int idxMin = i;
        for (int j = i+1; j < dataDosen05.length; j++) {
            if (dataDosen05 [j].usia > dataDosen05
[idxMin].usia) {
                idxMin = j;
            }
        }
        Dosen05 temp = dataDosen05 [idxMin];
        dataDosen05 [idxMin] = dataDosen05 [i];
        dataDosen05 [i] = temp;
    }
}

void insertionSort () {
    for (int i = 1; i < dataDosen05.length; i++) {
        Dosen05 temp = dataDosen05 [i];
        int j = i;
        while (j > 0 && dataDosen05 [j-1].usia < temp.usia) {
            dataDosen05 [j] = dataDosen05 [j-1];
            j--;
        }
        dataDosen05 [j] = temp;
    }
}
}

```

Kode Program DosenDemo05.java

```
package Jobsheet6;
import java.util.Scanner;
public class DosenDemo05 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DataDosen05 dosen = new DataDosen05();
        for (int i = 0; i < 10; i++) {
            System.out.println("Masukkan Data Dosen Ke-" + (i + 1));
            System.out.print("Masukkan Kode                : ");
            String kode = sc.nextLine();
            System.out.print("Masukkan Nama                : ");
            String nama = sc.nextLine();
            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();
            Dosen05 dsn = new Dosen05(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): ");
        dosen.sortingASC();
        dosen.tampil();
    }
}
```

```

        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 menggunakan
INSERTION SORT (DSC) : ");

        dosen.insertionSort();

        dosen.tampil();

    }
}

```

Output

```

Masukkan Data Dosen Ke-1
Masukkan Kode           : 1
Masukkan Nama            : Athaulla
Jenis Kelamin (Pria/Wanita) : Pria
Masukkan Usia            : 19

Masukkan Data Dosen Ke-2
Masukkan Kode           : 2
Masukkan Nama            : Hafizh
Jenis Kelamin (Pria/Wanita) : Pria
Masukkan Usia            : 19

Masukkan Data Dosen Ke-3
Masukkan Kode           : 3
Masukkan Nama            : Yan
Jenis Kelamin (Pria/Wanita) : Pria
Masukkan Usia            : 34

Masukkan Data Dosen Ke-4
Masukkan Kode           : 4
Masukkan Nama            : Zahra
Jenis Kelamin (Pria/Wanita) : Wanita
Masukkan Usia            : 25

Masukkan Data Dosen Ke-5
Masukkan Kode           : 5
Masukkan Nama            : Budi
Jenis Kelamin (Pria/Wanita) : Pria
Masukkan Usia            : 55

```

```

Masukkan Data Dosen Ke-6
Masukkan Kode           : 6
Masukkan Nama            : Wicaksono
Jenis Kelamin (Pria/Wanita) : Pria
Masukkan Usia            : 45

Masukkan Data Dosen Ke-7
Masukkan Kode           : 7
Masukkan Nama            : Putri
Jenis Kelamin (Pria/Wanita) : Wanita
Masukkan Usia            : 33

Masukkan Data Dosen Ke-8
Masukkan Kode           : 8
Masukkan Nama            : Agnes
Jenis Kelamin (Pria/Wanita) : Wanita
Masukkan Usia            : 58

Masukkan Data Dosen Ke-9
Masukkan Kode           : 9
Masukkan Nama            : Inka
Jenis Kelamin (Pria/Wanita) : Wanita
Masukkan Usia            : 39

Masukkan Data Dosen Ke-10
Masukkan Kode           : 10
Masukkan Nama            : Chelsea
Jenis Kelamin (Pria/Wanita) : Wanita
Masukkan Usia            : 23

```


Data dosen sebelum sorting:

Kode Dosen : 1

Nama Dosen : Athaulla

Jenis Kelamin : Pria

Usia Dosen : 19

Kode Dosen : 2

Nama Dosen : Hafizh

Jenis Kelamin : Pria

Usia Dosen : 19

Kode Dosen : 3

Nama Dosen : Yan

Jenis Kelamin : Pria

Usia Dosen : 34

Kode Dosen : 4

Nama Dosen : Zahra

Jenis Kelamin : Wanita

Usia Dosen : 25

Kode Dosen : 5

Nama Dosen : Budi

Jenis Kelamin : Pria

Usia Dosen : 55

Kode Dosen : 6

Nama Dosen : Wicaksono

Jenis Kelamin : Pria

Usia Dosen : 45

Kode Dosen : 7

Nama Dosen : Putri

Jenis Kelamin : Wanita

Usia Dosen : 33

Kode Dosen : 8

Nama Dosen : Agnes

Jenis Kelamin : Wanita

Usia Dosen : 58

Kode Dosen : 9

Nama Dosen : Inka

Jenis Kelamin : Wanita

Usia Dosen : 39

Kode Dosen : 10

Nama Dosen : Chelsea

Jenis Kelamin : Wanita

Usia Dosen : 23

Data dosen setelah sorting menggunakan BUBBLE SORT (ASC):

Kode Dosen : 1

Nama Dosen : Athaulla

Jenis Kelamin : Pria

Usia Dosen : 19

Kode Dosen : 2

Nama Dosen : Hafizh

Jenis Kelamin : Pria

Usia Dosen : 19

```
Kode Dosen : 10
Nama Dosen : Chelsea
Jenis Kelamin : Wanita
Usia Dosen : 23
-----
Kode Dosen : 4
Nama Dosen : Zahra
Jenis Kelamin : Wanita
Usia Dosen : 25
-----
Kode Dosen : 7
Nama Dosen : Putri
Jenis Kelamin : Wanita
Usia Dosen : 33
-----
Kode Dosen : 3
Nama Dosen : Yan
Jenis Kelamin : Pria
Usia Dosen : 34
-----
Kode Dosen : 9
Nama Dosen : Inka
Jenis Kelamin : Wanita
Usia Dosen : 39
-----
Kode Dosen : 6
Nama Dosen : Wicaksono
Jenis Kelamin : Pria
Usia Dosen : 45
-----
Kode Dosen : 5
Nama Dosen : Budi
Jenis Kelamin : Pria
Usia Dosen : 55
```

```
Kode Dosen : 8
Nama Dosen : Agnes
Jenis Kelamin : Wanita
Usia Dosen : 58
-----

Data dosen setelah sorting menggunakan SELECTION SORT (DSC) :
Kode Dosen : 8
Nama Dosen : Agnes
Jenis Kelamin : Wanita
Usia Dosen : 58
-----
Kode Dosen : 5
Nama Dosen : Budi
Jenis Kelamin : Pria
Usia Dosen : 55
-----
Kode Dosen : 6
Nama Dosen : Wicaksono
Jenis Kelamin : Pria
Usia Dosen : 45
-----
Kode Dosen : 9
Nama Dosen : Inka
Jenis Kelamin : Wanita
Usia Dosen : 39
-----
Kode Dosen : 3
Nama Dosen : Yan
Jenis Kelamin : Pria
Usia Dosen : 34
-----
```

```
Kode Dosen : 7
Nama Dosen : Putri
Jenis Kelamin : Wanita
Usia Dosen : 33
-----
Kode Dosen : 4
Nama Dosen : Zahra
Jenis Kelamin : Wanita
Usia Dosen : 25
-----
Kode Dosen : 10
Nama Dosen : Chelsea
Jenis Kelamin : Wanita
Usia Dosen : 23
-----
Kode Dosen : 2
Nama Dosen : Hafizh
Jenis Kelamin : Pria
Usia Dosen : 19
-----
Kode Dosen : 1
Nama Dosen : Athaula
Jenis Kelamin : Pria
Usia Dosen : 19
-----

Data dosen setelah sorting menggunakan INSERTION SORT (DSC) :
Kode Dosen : 8
Nama Dosen : Agnes
Jenis Kelamin : Wanita
Usia Dosen : 58
-----
```

```
Kode Dosen : 5
Nama Dosen : Budi
Jenis Kelamin : Pria
Usia Dosen : 55
-----
Kode Dosen : 6
Nama Dosen : Wicaksono
Jenis Kelamin : Pria
Usia Dosen : 45
-----
Kode Dosen : 9
Nama Dosen : Inka
Jenis Kelamin : Wanita
Usia Dosen : 39
-----
Kode Dosen : 3
Nama Dosen : Yan
Jenis Kelamin : Pria
Usia Dosen : 34
-----
Kode Dosen : 7
Nama Dosen : Putri
Jenis Kelamin : Wanita
Usia Dosen : 33
-----
Kode Dosen : 4
Nama Dosen : Zahra
Jenis Kelamin : Wanita
Usia Dosen : 25
-----
Kode Dosen : 10
Nama Dosen : Chelsea
Jenis Kelamin : Wanita
Usia Dosen : 23
-----
```

Kode Dosen : 2
Nama Dosen : Hafizh
Jenis Kelamin : Pria
Usia Dosen : 19

Kode Dosen : 1
Nama Dosen : Athaulla
Jenis Kelamin : Pria
Usia Dosen : 19
