

JOBSHEET 6
LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN STRUKTUR
DATA



MUHAMMAD DAFFI FIROS ZAIDAN

244107020182

TI 1E

PROGRAM STUDI D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2025

PERCOBAAN 1

Class

```
public class Sorting17 {
    int[] data;
    int jumData;

    Sorting17(int Data[], int jmlDat) {
        jumData = jmlDat;
        data = new int[jmlDat];
        for (int i = 0; i < jumData; i++) {
            data[i] = Data[i];
        }
    }

    void bubbleSort() {
        int temp;
        for (int i = 0; i < jumData - 1; 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 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;
        }
    }

    void insertionSort() {
        for (int i = 1; i < jumData; 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;
        }
    }

    void tampil() {
        for (int i = 0; i < jumData; i++) {
            System.out.print(data[i] + " ");
        }
        System.out.println();
    }
}
```

Main

```
public class SortingMain17 {
    public static void main(String[] args) {
        int[] a = {20, 10, 2, 7, 12};
        Sorting17 dataurut1 = new Sorting17(a, a.length);
        dataurut1.bubbleSort();
        System.out.println("Data Sudah Diurutkan dengan BUBBLE SORT (ASC)");
        dataurut1.tampil();
        System.out.println();

        int[] b = {30, 20, 2, 8, 14};
        Sorting17 dataurut2 = new Sorting17(b, b.length);
        dataurut2.selectionSort();
        System.out.println("Data Sudah Diurutkan dengan SELECTION SORT (ASC)");
        dataurut2.tampil();
        System.out.println();

        int[] c1 = {40, 10, 5, 3, 2};
        Sorting17 dataurut3 = new Sorting17(c1, c1.length);
        dataurut3.insertionSort();
        System.out.println("Data Sudah Diurutkan dengan INSERTION SORT (ASC)");
        dataurut3.tampil();
    }
}
```

Hasil Percobaan

```
Data Sudah Diurutkan dengan BUBBLE SORT (ASC)
2 7 10 12 20

Data Sudah Diurutkan dengan SELECTION SORT (ASC)
2 8 14 20 30

Data Sudah Diurutkan dengan INSERTION SORT (ASC)
2 3 5 10 40
PS D:\Tugas Kuliah\SEMESTER 2\ASLD\Praktikum ASD\Praktikum-ASD>
```

PERTANYAAN

1. Kode tersebut membandingkan dua elemen bersebelahan dalam array. Jika elemen sebelumnya (data[j-1]) lebih besar daripada elemen saat ini (data[j]), maka keduanya ditukar. Ini bertujuan untuk mengurutkan data dari kecil ke besar.
2. Kode Program

```
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;
        }
    }
}
```

3. Kondisi while (j >= 0 && data[j] > temp) memastikan dua hal:

- Indeks j tidak kurang dari nol (masih dalam batas array).
- Nilai data[j] masih lebih besar daripada elemen yang akan disisipkan (temp).

Selama kedua syarat itu terpenuhi, elemen di posisi j akan digeser ke kanan, memberi ruang agar temp dapat ditempatkan pada posisi yang tepat di dalam urutan.

4. Perintah `data[j + 1] = data[j]` berfungsi untuk menggeser elemen di posisi `j` ke posisi `j + 1`, sehingga terbuka ruang bagi elemen yang sedang diinsert (`temp`) untuk ditempatkan pada posisi yang benar.

PERCOBAAN 2 -BUBBLE SORT

Class Mahasiswa

```
public class Mahasiswa17 {
    String nim;
    String nama;
    String kelas;
    double ipk;

    Mahasiswa17() {
    }

    public Mahasiswa17(String rm, String name, String kls, double ip) {
        nim = rm;
        nama = name;
        kelas = kls;
        ipk = ip;
    }

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

Class MahasiswaBerprestasi

```
public class MahasiswaBerprestasi17 {
    Mahasiswa17[] listMhs = new Mahasiswa17[5];
    int idx;

    public MahasiswaBerprestasi17() {
    }

    void tambah(Mahasiswa17 m) {
        if (idx < listMhs.length) {
            listMhs[idx] = m;
            idx++;
        } else {
            System.out.println("Data sudah penuh!!");
        }
    }

    void tampil() {
        for (int i = 0; i < idx; i++) {
            listMhs[i].tampilkanInformasi();
        }
    }

    void bubbleSort() {
        for (int i = 0; i < idx - 1; i++) {
            for (int j = 1; j < idx - i; j++) {
                if (listMhs[j].ipk < listMhs[j - 1].ipk) {
                    Mahasiswa17 temp = listMhs[j];
                    listMhs[j] = listMhs[j - 1];
                    listMhs[j - 1] = temp;
                }
            }
        }
    }
}
```

MahasiswaDemo

```
public class MahasiswaDemo17 {  
    public static void main(String[] args) {  
        MahasiswaBerprestasi17 list = new MahasiswaBerprestasi17();  
  
        Mahasiswa17 m1 = new Mahasiswa17("123", "Zidan", "2A", 3.2);  
        Mahasiswa17 m2 = new Mahasiswa17("124", "Ayu", "2A", 3.5);  
        Mahasiswa17 m3 = new Mahasiswa17("125", "Sofi", "2A", 3.1);  
        Mahasiswa17 m4 = new Mahasiswa17("126", "Sita", "2A", 3.9);  
        Mahasiswa17 m5 = new Mahasiswa17("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 Sesudah Sorting Berdasarkan IPK (DESC)");  
        list.bubbleSort();  
    }  
}
```

Hasil Percobaan

```
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 Sesudah 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. Jawab Pertanyaan A, B, C

a. **Syarat dari perulangan adalah $i < \text{listMhs.length} - 1$?**

Syarat ini memastikan bahwa indeks i tidak melebihi panjang array dikurangi satu. Ini penting untuk menghindari kesalahan saat mengakses elemen yang tidak ada.

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

Syarat ini digunakan untuk mengurangi jumlah perbandingan yang dilakukan pada elemen yang sudah terurut. Setiap iterasi i , elemen terakhir tidak perlu dibandingkan lagi.

c. **Jika banyak data dalam listMhs adalah 50, maka berapakah perulangan yang akan berlangsung? Dan ada berapa tahap bubble sort yang ditempuh?**

Jika $\text{listMhs.length} = 50$:

Perulangan i berlangsung 49 kali.

Bubble Sort menempuh 49 tahap.

Total perbandingan dalam proses: 1225 kali.

2. Modifikasi

Class MahasiswaBerprestasi

```
public class MahasiswaBerprestasi17 {
    Mahasiswa17[] listMhs = new Mahasiswa17[5];
    int idx;

    void tambah(Mahasiswa17 m) {
        if (idx < listMhs.length) {
            listMhs[idx] = m;
            idx++;
        } else {
            System.out.println("Data sudah penuh");
        }
    }

    void tampil() {
        for (Mahasiswa17 m : listMhs) {
            if (m != null) {
                m.tampilkanInformasi();
            }
        }
    }

    void bubbleSort() {
        for (int i = 0; i < listMhs.length - 1; i++) {
            for (int j = 1; j < listMhs.length - i; j++) {
                if (listMhs[j] != null && listMhs[j - 1] != null) { //
                    Check null to avoid error
                    if (listMhs[j].ipk < listMhs[j - 1].ipk) {
                        Mahasiswa17 tmp = listMhs[j];
                        listMhs[j] = listMhs[j - 1];
                        listMhs[j - 1] = tmp;
                    }
                }
            }
        }
    }
}
```

Class MahasiswaDemo

```
import java.util.Scanner;

public class MahasiswaDemo17 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        MahasiswaBerprestasi17 list = new MahasiswaBerprestasi17();

        System.out.println("Data Mahasiswa Sebelum Sorting:");
        list.tampil();

        for (int i = 0; i < 5; 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 ipkStr = sc.nextLine().replace(",", ".");
            double ipk = Double.parseDouble(ipkStr);

            Mahasiswa17 m = new Mahasiswa17(nim, nama, kelas, ipk);
            list.tambah(m);

            System.out.println("-----");
        }

        System.out.println("\nData Mahasiswa Setelah Sorting Berdasarkan IPK (DESC):");
        list.bubbleSort();
        list.tampil();
    }
}
```

Hasil Kode Program

```
Data Mahasiswa Sebelum Sorting:
Masukkan Data Mahasiswa ke-1
NIM : 123
Nama : Ayu
Kelas : 2A
IPK : 3.1
-----
Masukkan Data Mahasiswa ke-2
NIM : 124
Nama : Daffi
Kelas : 2A
IPK : 4.0
-----
Masukkan Data Mahasiswa ke-3
NIM : 125
Nama : Zaidan
Kelas : 2A
IPK : 3.1
-----
Masukkan Data Mahasiswa ke-4
NIM : 156
Nama : Firos
Kelas : 2A
IPK : 3.5
-----
Masukkan Data Mahasiswa ke-5
NIM : 1A6
Nama : Muhammad
Kelas : 2A
IPK : 2.9
-----
```

```

Data Mahasiswa Setelah Sorting Berdasarkan IPK (DESC):
Nama : Muhammad
NIM : 146
Kelas : 2A
IPK : 2.9
Nama : Ayu
NIM : 123
Kelas : 2A
IPK : 3.1
Nama : Zaidan
NIM : 125
Kelas : 2A
IPK : 3.1
Nama : Firos
NIM : 156
Kelas : 2A
IPK : 3.5
Nama : Daffi
NIM : 124
Kelas : 2A
IPK : 4.0
PS D:\Tugas Kuliah\SEMESTER 2\ASLD\Praktikum ASD\Praktikum-ASD>

```

-SELECTION SORT

Class MahasiswaBerprestasi

```

public class MahasiswaBerprestasi17 {
    Mahasiswa17[] listMhs = new Mahasiswa17[5];
    int idx;

    void tambah(Mahasiswa17 m) {
        if (idx < listMhs.length) {
            listMhs[idx] = m;
            idx++;
        } else {
            System.out.println("Data sudah penuh");
        }
    }

    void tampil() {
        for (Mahasiswa17 m : listMhs) {
            if (m != null) {
                m.tampilkanInformasi();
            }
        }
    }

    void bubbleSort() {
        for (int i = 0; i < listMhs.length - 1; i++) {
            for (int j = 1; j < listMhs.length - i; j++) {
                if (listMhs[j] != null && listMhs[j - 1] != null) {
                    if (listMhs[j].ipk > listMhs[j - 1].ipk) {
                        Mahasiswa17 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;
            for (int j = i + 1; j < listMhs.length; j++) {
                if (listMhs[j] != null && listMhs[idxMin] != null) {
                    if (listMhs[j].ipk < listMhs[idxMin].ipk) {
                        idxMin = j;
                    }
                }
            }
            Mahasiswa17 tmp = listMhs[idxMin];
            listMhs[idxMin] = listMhs[i];
            listMhs[i] = tmp;
        }
    }
}

```


Class MahasiswaDemo

```
import java.util.Scanner;

public class MahasiswaDemo17 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        MahasiswaBerprestasi17 list = new MahasiswaBerprestasi17();

        System.out.println("Data Mahasiswa Sebelum Sorting:");
        list.tampil();

        for (int i = 0; i < 5; 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 ipkStr = sc.nextLine().replace(",", ".");
            double ipk = Double.parseDouble(ipkStr);

            Mahasiswa17 m = new Mahasiswa17(nim, nama, kelas, ipk);
            list.tambah(m);

            System.out.println("-----");
        }

        System.out.println("\nData Mahasiswa Setelah Sorting Berdasarkan IPK (DESC):");
        list.selectionSort();
        list.tampil();
    }
}
```

Hasil Kode Program

```
Data Mahasiswa Sebelum Sorting:
Masukkan Data Mahasiswa ke-1
NIM : 123
Nama : Ali
Kelas : 2B
IPK : 3.9
-----
Masukkan Data Mahasiswa ke-2
NIM : 124
Nama : ila
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 Mahasiswa Setelah Sorting Berdasarkan IPK (DESC):
Nama : ila
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

1. Baris program ini digunakan dalam proses *selection sort* untuk menemukan indeks elemen terkecil dari array yang belum diurutkan.

Variabel:

``idxMin`` untuk Menyimpan indeks elemen terkecil. Dimulai dengan nilai `i`, yang merupakan titik awal dari iterasi saat ini.

``j`` untuk Variabel pengulangan yang digunakan untuk memeriksa elemen-elemen dalam array.

- Proses ini bertujuan untuk menemukan elemen terkecil dalam subarray yang belum terurut, sehingga dapat ditukar dengan elemen di posisi ``i`` setelah loop selesai. Hal ini secara bertahap mengurutkan elemen-elemen dalam daftar dari yang terkecil hingga terbesar.

-INSERTION SORT

Class MahasiswaBerprestasi

```

public class MahasiswaBerprestasi17 {
    Mahasiswa17[] listMhs = new Mahasiswa17[5];
    int idx;

    void tambah(Mahasiswa17 m) {
        if (idx < listMhs.length) {
            listMhs[idx] = m;
            idx++;
        } else {
            System.out.println("Data sudah penuh");
        }
    }

    void tampil() {
        for (Mahasiswa17 m : listMhs) {
            if (m != null) {
                m.tampilkanInformasi();
            }
        }
    }

    void bubbleSort() {
        for (int i = 0; i < listMhs.length - 1; i++) {
            for (int j = 1; j < listMhs.length - i; j++) {
                if (listMhs[j] != null && listMhs[j - 1] != null) {
                    if (listMhs[j].ipk > listMhs[j - 1].ipk) {
                        Mahasiswa17 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;
        for (int j = i + 1; j < listMhs.length; j++) {
            if (listMhs[j] != null && listMhs[idxMin] != null) {
                if (listMhs[j].ipk < listMhs[idxMin].ipk) {
                    idxMin = j;
                }
            }
        }
        Mahasiswa17 tmp = listMhs[idxMin];
        listMhs[idxMin] = listMhs[i];
        listMhs[i] = tmp;
    }
}

void insertionSort() {
    for (int i = 1; i < listMhs.length; i++) {
        Mahasiswa17 temp = listMhs[i];
        int j = i;
        while (j > 0 && listMhs[j - 1] != null && temp != null &&
            listMhs[j - 1].ipk > temp.ipk) {
            listMhs[j] = listMhs[j - 1];
            j--;
        }
        listMhs[j] = temp;
    }
}
}

```

Hasil

```

Data Mahasiswa Sebelum Sorting:
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 : 3.0
-----
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 Mahasiswa Setelah Sorting Berdasarkan IPK (DESC):
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
PS D:\Tugas Kuliah\SEMESTER 2\ASLD\Praktikum ASD\Praktikum-ASD
> |
```

PERTANYAAN

```
void insertionSort() {
    for (int i = 1; i < listMhs.length; i++) {
        Mahasiswa17 temp = listMhs[i];
        int j = i;

        // Perbaiki indeks dari j1 ke j - 1
        while (j > 0 && listMhs[j - 1].ipk > temp.ipk) {
            listMhs[j] = listMhs[j - 1];
            j--;
        }

        listMhs[j] = temp;
    }
}
```

Hasil

```
Data Mahasiswa Sebelum Sorting:
Masukkan Data Mahasiswa ke-1
NIM : 111
Nama : ayu
Kelas : 2c
IPK : 3.7
```

```
-----
Masukkan Data Mahasiswa ke-2
NIM : 222
Nama : dika
Kelas : 2.c
IPK : 3.0
```

```
-----
Masukkan Data Mahasiswa ke-3
NIM : 333
Nama : ila
Kelas : 2c
IPK : 3.0
```

```
-----
Masukkan Data Mahasiswa ke-4
NIM : 444
Nama : yayuk
Kelas : 2c
IPK : 3.1
```

```
-----
Masukkan Data Mahasiswa ke-5
NIM : 555
Nama : susi
Kelas : 2c
IPK : 3.4
```

```
-----
Data Mahasiswa Setelah Sorting Berdasarkan IPK (DESC):
```

```
Nama : dika
NIM : 222
Kelas : 2.c
IPK : 3.0
Nama : ila
NIM : 333
Kelas : 2c
IPK : 3.0
Nama : yayuk
NIM : 444
Kelas : 2c
IPK : 3.1
Nama : susi
NIM : 555
Kelas : 2c
IPK : 3.4
Nama : ayu
NIM : 111
Kelas : 2c
IPK : 3.7
```

```
PS D:\Tugas Kuliah\SEMESTER 2\ASLD\Praktikum ASD\Praktikum-ASD> █
```

LATIHAN

Class Dosen

```
public class Dosen17 {
    String kode;
    String nama;
    boolean jenisKelamin;
    int usia;

    public Dosen17(String kodeDosen, String namaDosen, boolean kelamin, int umur) {
        this.kode = kodeDosen;
        this.nama = namaDosen;
        this.jenisKelamin = kelamin;
        this.usia = umur;
    }

    public void tampilkanInfo() {
        System.out.println("\nInformasi Dosen");
        System.out.println("Kode Dosen      : " + kode);
        System.out.println("Nama Dosen      : " + nama);
        System.out.println("Jenis Kelamin   : " + (jenisKelamin ? "Laki-laki" :
"Perempuan"));
        System.out.println("Usia            : " + usia + " tahun");
        System.out.println("=====");
    }
}
```

Class DataDosen

```
public class DataDosen17 {
    Dosen17[] daftarDosen = new Dosen17[10];
    int jumlah = 0;

    void tambahDosen(Dosen17 dosen) {
        if (jumlah < daftarDosen.length) {
            daftarDosen[jumlah] = dosen;
            jumlah++;
        } else {
            System.out.println("Kapasitas data dosen penuh!");
        }
    }

    void tampilkanSemua() {
        if (jumlah == 0) {
            System.out.println("Tidak ada data dosen.");
        } else {
            for (int i = 0; i < jumlah; i++) {
                daftarDosen[i].tampilkanInfo();
            }
        }
    }

    void sortingASC() {
        for (int i = 0; i < jumlah - 1; i++) {
            for (int j = 0; j < jumlah - 1 - i; j++) {
                if (daftarDosen[j].usia > daftarDosen[j + 1].usia) {
                    Dosen17 temp = daftarDosen[j];
                    daftarDosen[j] = daftarDosen[j + 1];
                    daftarDosen[j + 1] = temp;
                }
            }
        }
    }

    void sortingDSC() {
        for (int i = 0; i < jumlah - 1; i++) {
            int maxIndex = i;
            for (int j = i + 1; j < jumlah; j++) {
                if (daftarDosen[j].usia > daftarDosen[maxIndex].usia) {
                    maxIndex = j;
                }
            }
            Dosen17 temp = daftarDosen[maxIndex];
            daftarDosen[maxIndex] = daftarDosen[i];
            daftarDosen[i] = temp;
        }
    }
}
```

```

void insertionSort() {
    for (int i = 1; i < jumlah; i++) {
        Dosen17 temp = daftarDosen[i];
        int j = i;
        while (j > 0 && daftarDosen[j - 1].usia < temp.usia) {
            daftarDosen[j] = daftarDosen[j - 1];
            j--;
        }
        daftarDosen[j] = temp;
    }
}
}

```

Class DosenMain

```

import java.util.Scanner;

public class DosenMain17 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DataDosen17 listDosen = new DataDosen17();

        while (true) {
            System.out.println("===== Menu =====");
            System.out.println("1. Tambahkan Data Dosen");
            System.out.println("2. Tampilkan Data Dosen");
            System.out.println("3. Urutkan Data Bubble Sort (ASC)");
            System.out.println("4. Urutkan Data Selection Sort (DSC)");
            System.out.println("5. Urutkan Data Insertion Sort (DSC)");
            System.out.println("0. Keluar");
            System.out.print("Pilih menu: ");
            int pilihan = sc.nextInt();
            sc.nextLine();

            switch (pilihan) {
                case 1:
                    System.out.println();
                    for (int i = 0; i < 10; i++) {
                        System.out.print("Kode : ");
                        String kode = sc.nextLine();
                        System.out.print("Nama : ");
                        String nama = sc.nextLine();
                        System.out.print("Jenis Kelamin (L/P) : ");
                        char jkInput = sc.nextLine().charAt(0);
                        boolean jk = (jkInput == 'L' || jkInput == 'l');
                        System.out.print("Usia : ");
                        int usia = sc.nextInt();
                        sc.nextLine();
                        System.out.println();

                        Dosen17 dsn = new Dosen17(kode, nama, jk, usia);
                        listDosen.tambahDosen(dsn);
                    }
                    break;
                case 2:
                    System.out.println();
                    System.out.println("=== Data Keseluruhan ===");
                    listDosen.tampilkanSemua();
                    break;
                case 3:
                    System.out.println();
                    System.out.println("Data telah diurutkan (ASC - Bubble
Sort).\n");

```

```

case 3:
    System.out.println();
    System.out.println("Data telah diurutkan (ASC - Bubble Sort).\n");
    listDosen.sortingASC();
    listDosen.tampilkanSemua();
    break;
case 4:
    System.out.println();
    System.out.println("Data telah diurutkan (DSC - Selection Sort).\n");
    listDosen.sortingDSC();
    listDosen.tampilkanSemua();
    break;
case 5:
    System.out.println();
    System.out.println("Data telah diurutkan (DSC - Insertion Sort).\n");
    listDosen.insertionSort();
    listDosen.tampilkanSemua();
    break;
case 0:
    System.out.println("Keluar dari program.");
    sc.close();
    return;
default:
    System.out.println("Pilihan tidak valid.");
}
}
}
}
}

```

Hasil

```

===== MENU =====
1. Tambahkan Data Dosen
2. Tampilkan Data Dosen
3. Urutkan Data Bubble Sort (ASC)
4. Urutkan Data Selection Sort (DSC)
5. Urutkan Data Insertion Sort (ISC)
0. Keluar
Pilih menu: 1

Kode : 111
nama : Muhammad
Jenis Kelamin (L/P) : L
Usia : 22

Kode : 222
nama : Daffi
Jenis Kelamin (L/P) : L
Usia : 33

Kode : 333
nama : Firoz
Jenis Kelamin (L/P) : L
Usia : 30

Kode : 444
nama : Zaidan
Jenis Kelamin (L/P) : L
Usia : 43

Kode : 555
nama : Elay
Jenis Kelamin (L/P) : P
Usia : 22

Kode : 666
nama : Zafra
Jenis Kelamin (L/P) : P
Usia : 29

Kode : 777
nama : lllll
Jenis Kelamin (L/P) : P

```


Kode : 888
Nama : Udin
Jenis Kelamin (L/P) : L
Usia : 32

Kode : 999
Nama : Kisa
Jenis Kelamin (L/P) : P
Usia : 36

Kode : 000
Nama : Yiko
Jenis Kelamin (L/P) : P
Usia : 43

===== Menu =====

1. Tambahkan Data Dosen
2. Tampilkan Data Dosen
3. Urutkan Data Bubble Sort (ASC)
4. Urutkan Data Selection Sort (DSC)
5. Urutkan Data Insertion Sort (DSC)
0. Keluar

Pilih menu: 2

=== Data Keseluruhan ===

Informasi Dosen

Kode Dosen : 111
Nama Dosen : Muhammad
Jenis Kelamin : Laki-laki
Usia : 22 tahun

Informasi Dosen

Kode Dosen : 222
Nama Dosen : Daffi
Jenis Kelamin : Laki-laki
Usia : 33 tahun

Informasi Dosen

Kode Dosen : 333
Nama Dosen : Firos
Jenis Kelamin : Laki-laki
Usia : 39 tahun

Informasi Dosen

Kode Dosen : 444
Nama Dosen : Zaidan
Jenis Kelamin : Laki-laki
Usia : 43 tahun

Informasi Dosen

Kode Dosen : 555
Nama Dosen : Ilvy
Jenis Kelamin : Perempuan
Usia : 22 tahun

Informasi Dosen

Kode Dosen : 666
Nama Dosen : Zahra
Jenis Kelamin : Perempuan
Usia : 29 tahun

Informasi Dosen

Kode Dosen : 777
Nama Dosen : Lilis
Jenis Kelamin : Perempuan
Usia : 33 tahun

Informasi Dosen

Kode Dosen : 888
Nama Dosen : Udin
Jenis Kelamin : Laki-laki
Usia : 32 tahun

Informasi Dosen
Kode Dosen : 999
Nama Dosen : Kisa
Jenis Kelamin : Perempuan
Usia : 36 tahun
=====

Informasi Dosen
Kode Dosen : 000
Nama Dosen : Yiko
Jenis Kelamin : Perempuan
Usia : 43 tahun
=====

===== Menu =====

1. Tambahkan Data Dosen
 2. Tampilkan Data Dosen
 3. Urutkan Data Bubble Sort (ASC)
 4. Urutkan Data Selection Sort (DESC)
 5. Urutkan Data Insertion Sort (DESC)
 0. Keluar
- Pilih menu: 3

Data telah diurutkan (ASC - Bubble Sort).

Informasi Dosen
Kode Dosen : 111
Nama Dosen : Muhammad
Jenis Kelamin : Laki-laki
Usia : 22 tahun
=====

Informasi Dosen
Kode Dosen : 555
Nama Dosen : Ilvy
Jenis Kelamin : Perempuan
Usia : 22 tahun
=====

Informasi Dosen
Kode Dosen : 666
Nama Dosen : Zahra
=====

Informasi Dosen
Kode Dosen : 888
Nama Dosen : Udin
Jenis Kelamin : Laki-laki
Usia : 32 tahun
=====

Informasi Dosen
Kode Dosen : 222
Nama Dosen : Daffi
Jenis Kelamin : Laki-laki
Usia : 33 tahun
=====

Informasi Dosen
Kode Dosen : 777
Nama Dosen : Lili
Jenis Kelamin : Perempuan
Usia : 33 tahun
=====

Informasi Dosen
Kode Dosen : 999
Nama Dosen : Kisa
Jenis Kelamin : Perempuan
Usia : 36 tahun
=====

Informasi Dosen
Kode Dosen : 333
Nama Dosen : Firoz
Jenis Kelamin : Laki-laki
Usia : 39 tahun
=====

Informasi Dosen
Kode Dosen : 444
Nama Dosen : Zaidan
Jenis Kelamin : Laki-laki
Usia : 43 tahun
=====

```
Informasi Dosen
Kode Dosen   : 000
Nama Dosen   : Yiko
Jenis Kelamin : Perempuan
Usia         : 43 tahun
=====

----- Menu -----
1. Tambahkan Data Dosen
2. Tampilkan Data Dosen
3. Urutkan Data Bubble Sort (ASC)
4. Urutkan Data Selection Sort (DSC)
5. Urutkan Data Insertion Sort (DSC)
0. Keluar
Pilih menu: 4

Data telah diurutkan (DSC - Selection Sort).

Informasi Dosen
Kode Dosen   : 444
Nama Dosen   : Zaidan
Jenis Kelamin : Laki-laki
Usia         : 43 tahun
=====

Informasi Dosen
Kode Dosen   : 000
Nama Dosen   : Yiko
Jenis Kelamin : Perempuan
Usia         : 43 tahun
=====

Informasi Dosen
Kode Dosen   : 333
Nama Dosen   : Firoz
Jenis Kelamin : Laki-laki
Usia         : 39 tahun
=====

Informasi Dosen
Kode Dosen   : 999
Nama Dosen   : Kisa
Informasi Dosen
Kode Dosen   : 222
Nama Dosen   : Daffi
Jenis Kelamin : Laki-laki
Usia         : 33 tahun
=====

Informasi Dosen
Kode Dosen   : 777
Nama Dosen   : Lilis
Jenis Kelamin : Perempuan
Usia         : 33 tahun
=====

Informasi Dosen
Kode Dosen   : 888
Nama Dosen   : Udin
Jenis Kelamin : Laki-laki
Usia         : 32 tahun
=====

Informasi Dosen
Kode Dosen   : 666
Nama Dosen   : Zahra
Jenis Kelamin : Perempuan
Usia         : 29 tahun
=====

Informasi Dosen
Kode Dosen   : 111
Nama Dosen   : Muhammad
Jenis Kelamin : Laki-laki
Usia         : 22 tahun
=====

Informasi Dosen
Kode Dosen   : 555
Nama Dosen   : Ilvy
Jenis Kelamin : Perempuan
Usia         : 22 tahun
=====
```

```

===== Menu =====
1. Tambahkan Data Dosen
2. Tampilkan Data Dosen
3. Urutkan Data Bubble Sort (ASC)
4. Urutkan Data Selection Sort (DESC)
5. Urutkan Data Insertion Sort (DESC)
0. Keluar
Pilih menu: 5

Data telah diurutkan (DESC = Insertion Sort).

Informasi Dosen
Kode Dosen      : 444
Nama Dosen      : Zaidan
Jenis Kelamin    : Laki-laki
Usia             : 43 tahun
=====

Informasi Dosen
Kode Dosen      : 888
Nama Dosen      : Yiko
Jenis Kelamin    : Perempuan
Usia             : 43 tahun
=====

Informasi Dosen
Kode Dosen      : 333
Nama Dosen      : Firoz
Jenis Kelamin    : Laki-laki
Usia             : 39 tahun
=====

Informasi Dosen
Kode Dosen      : 999
Nama Dosen      : Kisa
Jenis Kelamin    : Perempuan
Usia             : 36 tahun
=====

Informasi Dosen
Kode Dosen      : 222
Informasi Dosen
Kode Dosen      : 222
Nama Dosen      : Daffi
Jenis Kelamin    : Laki-laki
Usia             : 33 tahun
=====

Informasi Dosen
Kode Dosen      : 777
Nama Dosen      : Lilis
Jenis Kelamin    : Perempuan
Usia             : 33 tahun
=====

Informasi Dosen
Kode Dosen      : 888
Nama Dosen      : Udin
Jenis Kelamin    : Laki-laki
Usia             : 32 tahun
=====

Informasi Dosen
Kode Dosen      : 666
Nama Dosen      : Zahra
Jenis Kelamin    : Perempuan
Usia             : 29 tahun
=====

Informasi Dosen
Kode Dosen      : 111
Nama Dosen      : Muhammad
Jenis Kelamin    : Laki-laki
Usia             : 22 tahun
=====

Informasi Dosen
Kode Dosen      : 555
Nama Dosen      : Ilvy
Jenis Kelamin    : Perempuan
Usia             : 22 tahun
=====

```

===== Menu =====

1. Tambahkan Data Dosen
2. Tampilkan Data Dosen
3. Urutkan Data Bubble Sort (ASC)
4. Urutkan Data Selection Sort (DSC)
5. Urutkan Data Insertion Sort (DSC)
0. Keluar

Pilih menu: 0

Keluar dari program.

PS D:\Tugas Kuliah\SEMESTER 2\ASLD\Praktikum ASD\Praktikum-ASD>