

LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA
JOBSHEET 6



RIFO ANGGI BARBARA DANUARTA
244107020063
TI_1E
PROGRAM STUDI D_IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

Percobaan 1

a. SORTING – BUBBLE SORT

1. Buat folder baru bernama Jobsheet6 di dalam repository Praktikum ASD
2. Buat class Sorting, kemudian tambahkan atribut sebagai berikut:

```
public class Sorting26 {  
  
    int [] data;  
    int jumData;
```

3. Buatlah konstruktor dengan parameter Data[] dan jmlDat

```
Sorting26 (int Data[], int jmlDat) {  
    jumData = jmlDat;  
    data = new int[jmlDat];  
    for (int i = 0; i < jumData; i++) {  
        data[i] = Data[i];  
    }  
}
```

4. Buatlah method bubbleSort bertipe void dan deklarasikan isinya menggunakan algoritma Bubble Sort.

```
void bubbleSort() {  
    int temp = 0;  
    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;  
            }  
        }  
    }  
}
```

5. Buatlah method tampil bertipe void dan deklarasikan isi method tersebut.

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

6. Buat class SortingMain kemudian deklarasikan array dengan nama a[] kemudian isi array tersebut

```
int a[] = {20, 10, 2, 7, 12};
```

7. Buatlah objek baru dengan nama dataurut1 yang merupakan instansiasi dari class Sorting, kemudian isi parameternya

```
Sorting26 dataurut1 = new Sorting26(a, a.length);
```

8. Lakukan pemanggilan method bubbleSort dan tampil

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

9. Jalankan program, dan amati hasilnya!

```
PS E:\Semester 2\Algoritma dan Struktur Data\Praktikum-ASD>
.\jdk-23\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetails'
cp' 'C:\Users\THINKPAD\AppData\Roaming\Code\User\workspaceStorage
00ef0ddb9ec00\redhat.java\jdt_ws\Praktikum-ASD_3c4ba0fc\bin'
.
Data awal 1
20 10 2 7 12
Data sudah diurutkan dengan BUBBLE SORT (ASC)
2 7 10 12 20
PS E:\Semester 2\Algoritma dan Struktur Data\Praktikum-ASD>
```

b. SORTING – SELECTION SORT

1. Pada class Sorting yang sudah dibuat di praktikum sebelumnya tambahkan method SelectionSort yang mengimplementasikan pengurutan menggunakan algoritma selection sort.

```
void SelectionSort() {
    for (int i = 0; i < jumData; 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;
    }
}
```

2. Deklarasikan array dengan nama b[] pada kelas SortingMain kemudian isi array tersebut

```
int b[] = {30, 20, 2, 8, 14};
```

3. Buatlah objek baru dengan nama dataurut2 yang merupakan instansiasi dari class Sorting, kemudian isi parameternya

```
Sorting26 dataurut2 = new Sorting26(b, b.length);
```

4. Lakukan pemanggilan method SelectionSort dan tampil

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

5. Jalankan program dan amati hasilnya!

```
PS E:\Semester 2\Algoritma dan Struktur Data\Praktikum-ASD>
.\jdk-23\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetails'
cp' 'C:\Users\THINKPAD\AppData\Roaming\Code\User\workspaceStorage\
00ef0ddb9ec00\redhat.java\jdt_ws\Praktikum-ASD_3c4ba0fc\bin'
'

Data awal 1
20 10 2 7 12
Data sudah diurutkan dengan BUBBLE SORT (ASC)
2 7 10 12 20
Data awal 2
30 20 2 8 14
Data sudah diurutkan dengan SELECTION SORT (ASC)
2 8 14 20 30
PS E:\Semester 2\Algoritma dan Struktur Data\Praktikum-ASD>
```

c. SORTING – INSERTION SORT

1. Pada class Sorting yang sudah dibuat di praktikum sebelumnya tambahkan method insertionSort yang mengimplementasikan pengurutan menggunakan algoritma insertion sort.

```
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;
    }
}
```

2. Deklarasikan array dengan nama c[] pada kelas SortingMain kemudian isi array tersebut

```
int c[] = {40, 10, 4, 9, 3};
```

3. Buatlah objek baru dengan nama dataurut3 yang merupakan instansiasi dari class Sorting, kemudian isi parameternya

```
Sorting26 dataurut3 = new Sorting26(c, c.length);
```

4. Lakukan pemanggilan method insertionSort dan tampil

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

5. Jalankan program dan amati hasilnya!

```
PS E:\Semester 2\Algoritma dan Struktur Data\Praktikum-ASD>
.\jdk-23\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetails'
cp' 'C:\Users\THINKPAD\AppData\Roaming\Code\User\workspaceStorage\
00ef0ddb9ec00\redhat.java\jdt_ws\Praktikum-ASD_3c4ba0fc\bin'
'
Data awal 1
20 10 2 7 12
Data sudah diurutkan dengan BUBBLE SORT (ASC)
2 7 10 12 20
Data awal 2
30 20 2 8 14
Data sudah diurutkan dengan SELECTION SORT (ASC)
2 8 14 20 30
Data awal 3
40 10 4 9 3
Data sudah diurutkan dengan INSERTION SORT (ASC)
3 4 9 10 40
PS E:\Semester 2\Algoritma dan Struktur Data\Praktikum-ASD>
```

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;
```

2. Tunjukkan kode program yang merupakan algoritma pencarian nilai minimum pada selection sort!
3. Pada Insertion sort , jelaskan maksud dari kondisi pada perulangan
While (l >= 0 && data[i] > temp)
4. Pada Insertion sort, apakah tujuan dari perintah data[j + 1] = data[j];

Jawaban

1. if (data[j-1] > data[j]) = membandingkan dua elemen yang bersebelahan
temp = data[j]; = menyimpan nilai data[j] di temp
data[j] = data[j-1]; = gantikan data[j] menjadi data[j-1]
data[j-1] = temp; = memasukan nilai temp ke data[j-1]
2. for (int j = i + 1; j < jumData; j++) {
 if (data[j] < data[min]) {
 min = j;
 }
}
3. Menggeser elemen yang lebih besar ke kanan untuk memberi ruang bagi elemen baru.
Memastikan bahwa elemen yang lebih kecil ditempatkan di posisi yang benar dalam bagian array yang sudah diurutkan.
Terus berjalan sampai menemukan posisi yang benar atau mencapai batas kiri array (i = -1).
4. Yaitu menyimpan nilai dari data[j] ke data[j+1]

Percobaan 2

1. Buatlah class dengan nama Mahasiswa.
2. Untuk lebih jelasnya class tersebut dapat dilihat pada potongan kode di bawah ini

```
package Jobsheet6;

public class Mahasiswa26 {
    String nim;
    String nama;
    String kelas;
    double ipk;

    Mahasiswa26() {

    }

    Mahasiswa26 (String nm, String name, String kls, double ip
) {
        nim = nm;
        nama = name;
        kelas = kls;
        ipk = ip;

    }

    void tampilInformasi () {

        System.out.println("Nama: " + nama);
        System.out.println("Nim: " + nim);
        System.out.println("Kelas: " + kelas);
        System.out.println("Ipk: " + ipk);

    }

}
```

3. Buat class MahasiswaBerprestasi seperti di bawah ini!

```
public class MahasiswaBerprestasi26 {
    Mahasiswa26[] listMhs = new Mahasiswa26[5];
    int idx;
```

4. Tambahkan method tambah() di dalam class tersebut! Method tambah() digunakan untuk menambahkan objek dari class Mahasiswa ke dalam atribut listMhs.

```

void tambah (Mahasiswa26 m) {
    if (idx < listMhs.length) {
        listMhs[idx] = m;
        idx++;
    } else {
        System.out.println("data sudah penuh");
    }
}

```

5. Tambahkan method tampil() di dalam class tersebut! Method tampil() digunakan untuk menampilkan semua data mahasiswa-mahasiswa yang ada di dalam class tersebut! Perhatikan penggunaan sintaks for yang agak berbeda dengan for yang telah dipelajari sebelumnya, meskipun secara konsep sebenarnya mirip.

```

void tampil () {
    for (Mahasiswa26 m:listMhs) {
        m.tampilInformasi();
        System.out.println("-----");
    }
}

```

6. Tambahkan method bubbleSort() di dalam class tersebut!

```

void bubbleSort () {
    for (int i = 0; i < listMhs.length; i++) {
        for (int j = 1; j < listMhs.length; j++ ) {
            if (listMhs[j].ipk > listMhs[j-1].ipk) {
                Mahasiswa26 tmp = listMhs[j];
                listMhs[j] = listMhs[j-1];
                listMhs[j-1] = tmp;
            }
        }
    }
}

```

7. Buat class MahasiswaDemo, kemudian buatlah sebuah objek MahasiswaBerprestasi dan buatlah 5 objek mahasiswa kemudian tambahkan semua objek mahasiswa tersebut dengan memanggil fungsi tambah pada objek MahasiswaBerprestasi. Silakan dipanggil fungsi tampil() untuk melihat semua data yang telah dimasukan, urutkan data tersebut dengan memanggil fungsi bubbleSort() dan yang terakhir panggil fungsi tampil kembali.

```
public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    input.useLocale(Locale.US);

    MahasiswaBerprestasi26 list = new
MahasiswaBerprestasi26();

    System.out.print("Masukan Jumlah mahasiswa: ");
    int jlmMhs = input.nextInt();
    input.nextLine();

    for (int i = 0; i < jlmMhs; i++ ) {
        System.out.println("Masukan Data mahasiswa ke-" +
(i + 1));
        System.out.print("Nim: ");
        String nim = input.nextLine();
        System.out.print("Nama: ");
        String nama = input.nextLine();
        System.out.print("Kelas: ");
        String kelas = input.nextLine();
        System.out.print("IPK: ");
        double IPK = input.nextDouble();
        input.nextLine();

        Mahasiswa26 m = new Mahasiswa26(nim, nama, kelas,
IPK);
        list.tambah(m);
        System.out.println("-----
");
    }

    System.out.println("Data mahasiswa sebelum sorting: ");
    list.tampil();

    System.out.println("Data Mahasiswa Setelah sorting
berdasarkan IPK (DESC): ");
    list.bubbleSort();
    list.tampil();
}
```


8. Cocokkan hasilnya dengan yang terdapat pada tampilan di bawah ini

```
PS E:\Semester 2\Algoritma dan Struktur Data\Praktikum-ASD>
\THINKPAD\AppData\Roaming\Code\User\workspaceStorage\5a9b874
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
-----
PS E:\Semester 2\Algoritma dan Struktur Data\Praktikum-ASD>
```

Pertanyaan

1. Perhatikan perulangan di dalam bubbleSort() di bawah ini:
 - a. Mengapa syarat dari perulangan i adalah $i < \text{listMhs.length} - 1$?
 - b. Mengapa syarat dari perulangan j adalah $j < \text{listMhs.length} - i$?
 - c. Jika banyak data di dalam listMhs adalah 50, maka berapakah perulangan i akan berlangsung? Dan ada berapa Tahap bubble sort yang ditempuh?
2. Modifikasi program diatas dimana data mahasiswa bersifat dinamis (input dari keyborad) yang terdiri dari nim, nama, kelas, dan ipk!

Jawaban

1. A. Menghindari perbandingan yang tidak perlu, karena elemen terakhir otomatis benar.
B. melakukan perbandingan dan pertukaran elemen dalam array untuk menempatkan elemen terbesar ke posisi akhir pada setiap iterasi luar (i).
C. Perulangan i berlangsung sebanyak 49 kali
Bubble Sort menempuh 49 tahap untuk menyelesaikan pengurutan.

2.

Percobaan 3

1. Lihat kembali class MahasiswaBerprestasi, dan tambahkan method selectionSort() di dalamnya! Method ini juga akan melakukan proses sorting secara ascending, tetapi menggunakan pendekatan selection sort.

```
System.out.print("Masukan Jumlah mahasiswa: ");
int jlmMhs = input.nextInt();
input.nextLine();

for (int i = 0; i < jlmMhs; i++ ) {
    System.out.println("Masukan Data mahasiswa ke-" + (i +
1));
    System.out.print("Nim: ");
    String nim = input.nextLine();
    System.out.print("Nama: ");
    String nama = input.nextLine();
    System.out.print("Kelas: ");
    String kelas = input.nextLine();
    System.out.print("IPK: ");
    double IPK = input.nextDouble();
    input.nextLine();

    Mahasiswa26 m = new Mahasiswa26(nim, nama, kelas, IPK);
    list.tambah(m);
    System.out.println("-----");
}
```

- Setelah itu, buka kembali class MahasiswaDemo, dan di dalam method main() tambahkan baris program untuk memanggil method selectionSort() tersebut, kemudian panggil method tampil() untuk menampilkan data yang sudah diurutkan!

```
System.out.println("Data yang sudah terurut menggunakan  
SELECTION SORT (ASC)");  
list.SelectionSort();  
list.tampil();
```

- Coba jalankan kembali class MahasiswaDemo, dan amati hasilnya! Apakah kini data mahasiswa telah tampil urut menaik berdasar ipk?

```
PS E:\Semester 2\Algoritma dan Struktur Data\Praktikum-ASD>  
.\jdk-23\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetail'  
cp' 'C:\Users\THINKPAD\AppData\Roaming\Code\User\workspaceSt  
00ef0ddb9ec00\redhat.java\jdt_ws\Praktikum-ASD_3c4ba0fc\bin'  
26'  
Masukan Jumlah mahasiswa: 5  
Masukan Data mahasiswa ke-1  
Nim: 123  
Nama: ali  
Kelas: 2B  
IPK: 3.9  
-----  
Masukan Data mahasiswa ke-2  
Nim: 124  
Nama: ila  
Kelas: 2B  
IPK: 3.1  
-----  
Masukan Data mahasiswa ke-3  
Nim: 125  
Nama: agus  
Kelas: 2B  
IPK: 3.6  
-----  
Masukan Data mahasiswa ke-4  
Nim: 126  
Nama: tika  
Kelas: 2B  
IPK: 3.3  
-----  
Masukan Data mahasiswa ke-5  
Nim: 127  
Nama: udin  
Kelas: 2B  
IPK: 3.2  
-----  
Data yang sudah terurut menggunakan SELECTION SORT (ASC)  
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

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!

Jawaban

int idxMin = i; = mengasumsi bahwa elemen pertama adalah yang terkecil

Loop for (int j = i+1; j < listMhs.length; j++)= mengecek dari elemen pertama sampai akhir

if (listMhs[j].ipk < listMhs[idxMin].ipk) = jika menemukan yang lebih kecil akan update ke idxMin.

Percobaan 4

1. Lihat kembali class MahasiswaBerprestasi, dan tambahkan method insertionSort() di dalamnya. Method ini juga akan melakukan proses sorting secara ascending, tetapi menggunakan pendekatan Insertion Sort.

```
void insertionSort () {
    for (int i = 0; i < listMhs.length; i++) {
        Mahasiswa26 temp = listMhs[i];
        int j = i;
        while (j>0 && listMhs[j-1].ipk > temp.ipk) {
            listMhs[j]=listMhs[j-1];
            j--;
        }
        listMhs[j]=temp;
    }
}
```

2. Setelah itu, buka kembali class MahasiswaDemo, dan di dalam method main() tambahkan baris program untuk memanggil method insertionSort() dan tampil () tersebut!

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

3. Coba jalankan kembali class MahasiswaDemo, dan amati hasilnya! Apakah kini data mahasiswa telah tampil urut menaik berdasar ipk?

```
PS E:\Semester 2\Algoritma dan Struktur
\jdk-23\bin\java.exe' '--enable-preview'
cp' 'C:\Users\THINKPAD\AppData\Roaming\Co
00ef0ddb9ec00\redhat.java\jdt_ws\Praktik
26'
Masukan Jumlah mahasiswa: 5
Masukan Data mahasiswa ke-1
Nim: 111
Nama: ayu
Kelas: 2c
IPK: 3.7
-----
Masukan Data mahasiswa ke-2
Nim: 222
Nama: dika
Kelas: 2c
IPK: 3.0
-----
Masukan Data mahasiswa ke-3
Nim: 333
Nama: ila
Kelas: 2c
IPK: 3.8
-----
Masukan Data mahasiswa ke-4
Nim: 444
Nama: susi
Kelas: 2c
IPK: 3.1
-----
Masukan 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: susi
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 E:\Semester 2\Algoritma dan Struktur Data\Praktikum-ASD>
```

Pertanyaan

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

Jawaban

```
void insertionSort () {
    for (int i = 0; i < listMhs.length; i++) {
        Mahasiswa26 temp = listMhs[i];
        int j = i;
        while (j>0 && listMhs[j-1].ipk < temp.ipk) {
            listMhs[j]=listMhs[j-1];
            j--;
        }
        listMhs[j]=temp;
    }
}
```

Cukup diubah dari “listMhs[j-1].ipk < temp.ipk” ke “listMhs[j-1].ipk > temp.ipk”
Yaitu dari terbesar ke terkecil dirubah ke terkecil ke terbesar

Latihan

kode program

```
package Jobsheet6;

public class Dosen26 {
    String kode;
    String nama;
    Boolean jenis;
    int usia;

    Dosen26() {

    }

    Dosen26(String kd, String name, Boolean jk, int age) {
        kode = kd;
        nama = name;
        jenis = jk;
        usia = age;
    }

    void tampil () {
        String jns = jenis ? "Laki-Laki" : "Perempuan" ;
        System.out.println("Kode: " + kode);
        System.out.println("Nama: " + nama);
        System.out.println("Jenis: " + jenis);
        System.out.println("Usia: " + usia);
    }
}
```

```

package Jobsheet6;

public class DataDosen {
    Dosen26[] dataDosen = new Dosen26[10];
    int idx = 0;

    void tambah(Dosen26 dsn) {
        if (idx < dataDosen.length) {
            dataDosen[idx] = dsn;
            idx++;
        } else {
            System.out.println("Data sudah penuh!");
        }
    }

    void tampil() {
        if (idx == 0) {
            System.out.println("Belum ada data dosen.");
            return;
        }
        for (int i = 0; i < idx; i++) {
            if (dataDosen[i] != null) {
                dataDosen[i].tampil();
                System.out.println("-----");
            }
        }
    }

    void SortingASC() {
        for (int i = 0; i < idx - 1; i++) {
            for (int j = 1; j < idx - i; j++) {
                if (dataDosen[j].usia < dataDosen[j - 1].usia) {
                    Dosen26 tmp = dataDosen[j];
                    dataDosen[j] = dataDosen[j - 1];
                    dataDosen[j - 1] = tmp;
                }
            }
        }
    }

    void SortingDSC() {
        for (int i = 0; i < idx; i++) {
            Dosen26 temp = dataDosen[i];
            int j = i;
            while (j > 0 && dataDosen[j - 1].usia < temp.usia) {
                dataDosen[j] = dataDosen[j - 1];
                j--;
            }
            dataDosen[j] = temp;
        }
    }
}

```

Jobsheet6;

```
public class DosenMain {  
    public static void main(String[] args) {  
  
        DataDosen dataDosen = new DataDosen();  
  
        dataDosen.tambah(new Dosen26("D001", "Andi", true, 35));  
        dataDosen.tambah(new Dosen26("D002", "Siti", false, 40));  
        dataDosen.tambah(new Dosen26("D003", "Budi", true, 30));  
        dataDosen.tambah(new Dosen26("D004", "Ani", false, 45));  
  
        System.out.println("\nData Dosen sebelum disorting:");  
        dataDosen.tampil();  
  
        System.out.println("\nData telah diurutkan secara ASC  
berdasarkan usia:");  
        dataDosen.SortingASC();  
        dataDosen.tampil();  
  
        System.out.println("\nData telah diurutkan secara DSC  
berdasarkan usia:");  
        dataDosen.SortingDSC();  
        dataDosen.tampil();  
    }  
}
```


Output

```
PS E:\Semester 2\Algoritma dan Struktur Data\Praktikum-ASD>
sInExceptionMessages' '-cp' 'C:\Users\THINKPAD\AppData\Roami
tikum-ASD_3c4ba0fc\bin' 'Jobsheet6.DosenMain'
```

Data Dosen sebelum disorting:

Kode: D001

Nama: Andi

Jenis: true

Usia: 35

Kode: D002

Nama: Siti

Jenis: false

Usia: 40

Kode: D003

Nama: Budi

Jenis: true

Usia: 30

Kode: D004

Nama: Ani

Jenis: false

Usia: 45

Data telah diurutkan secara ASC berdasarkan usia:

Kode: D003

Nama: Budi

Jenis: true

Usia: 30

Kode: D001

Nama: Andi

Jenis: true

Usia: 35

Kode: D002

Nama: Siti

Jenis: false

Usia: 40

Kode: D004

Nama: Ani

Jenis: false

Usia: 45

```
Data telah diurutkan secara DSC berdasarkan usia:
Kode: D004
Nama: Ani
Jenis: false
Usia: 45
-----
Kode: D002
Nama: Siti
Jenis: false
Usia: 40
-----
Kode: D001
Nama: Andi
Jenis: true
Usia: 35
-----
Kode: D003
Nama: Budi
Jenis: true
Usia: 30
-----
PS E:\Semester 2\Algoritma dan Struktur Data\Praktikum-ASD>
```