

POLITEKNIK NEGERI MALANG
TEKNOLOGI INFORMASI
TEKNIK INFORMATIKA



Nama: Abdul Rahman Hanif Darmawan

NIM: 244107020232

Kelas: TI-1A

Prodi: D4-TEKNIK INFORMATIKA

JOBSHEET 5

SORTING (BUBBLE, SELECTION, DAN INSERTION SORT)

5.2.2 Verifikasi Hasil Percobaan

a. SORTING – BUBBLE SORT

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

b. SORTING – SELECTION SORT

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

c. SORTING – INSERTION SORT

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

5.2.5 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 tersebut merupakan bagian dari algoritma pertukaran elemen dalam proses sorting. Jika elemen sebelumnya data[j-1] lebih besar dari data[j], maka keduanya ditukar.

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

-

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

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

- Kondisi yang memastikan bahwa elemen yang lebih besar dari temp akan digeser ke kanan untuk memberi ruang bagi elemen yang diinsert.
4. Pada insertion sort, apakah tujuan dari perintah `data[j+1] = data[j];`
- Berfungsi untuk menggeser elemen ke kanan agar menciptakan ruang untuk elemen yang akan disisipkan (temp) di posisi yang sesuai dalam urutan yang telah diatur.

5.3.3 Verifikasi 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 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
-----

```

5.3.4 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`?
 - Karena pada setiap iterasi luar (i), elemen terbesar dalam sisa array akan dipindahkan ke posisinya yang benar.
- b. Mengapa syarat dari perulangan j adalah `j<listMhs.length-i` ?

- Karena setiap iterasi luar (i), elemen terbesar dari sisa array sudah berada di posisi yang benar, sehingga iterasi dalam (j) tidak perlu membandingkan elemen yang sudah terurut di akhir array.

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

- Perulangan i akan berlangsung sebanyak 49 kali ($50 - 1$).
- Setiap tahap i akan melakukan $50 - i$ perbandingan dalam loop j.
- Total jumlah perbandingan (kompleksitas) adalah $(50 \times 49) / 2 = 1225$ kali.

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

```
package Praktikum;
import java.util.Scanner;

public class MahasiswaDemo02 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Masukkan jumlah mahasiswa: ");
        int jumlah = input.nextInt();
        input.nextLine();

        MahasiswaBerprestasi02 list = new
        MahasiswaBerprestasi02(jumlah);

        for (int i = 0; i < jumlah; i++) {
            System.out.println("Masukkan 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();

            Mahasiswa02 mhs = new Mahasiswa02(nim, nama, kelas,
ipk);
            list.tambah(mhs);
        }

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

        System.out.println("\nData mahasiswa setelah sorting
berdasarkan IPK (DESC): ");
        list.bubbleSort();
        list.tampil();
    }
}

// Konstruktor agar bisa menampung jumlah mahasiswa ya
MahasiswaBerprestasi02(int jumlah) {
    listMhs = new Mahasiswa02[jumlah];
    idx = 0;
}
```

5.3.6 Verifikasi Hasil Percobaan

```
Masukkan jumlah mahasiswa: 5
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 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
-----
```

5.3.7 Pertanyaan

1. 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!

- Proses tersebut merupakan bagian dari Selection Sort yang berfungsi untuk mencari elemen dengan IPK terkecil dan menempatkannya di posisi yang benar pada setiap iterasi luar.

5.4.2 Verifikasi Hasil Percobaan

```
Masukkan jumlah mahasiswa: 5
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 : susi
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: susi
NIM: 444
Kelas: 2c
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
-----
```

5.4.3 Pertanyaan

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

-

```
void insertionSort(){
    for (int i = 1; i < listMhs.length; i++){
        Mahasiswa temp = listMhs[i];
        int j = i;
        while (j > 0 && listMhs[j-1].ipk < temp.ipk){ // Ubah tanda
            '<' untuk sorting descending
            listMhs[j] = listMhs[j-1];
            j--;
        }
        listMhs[j] = temp;
    }
}
```

5.5 Tugas

1. DosenMain02.java

```
package Tugas;
import java.util.Scanner;

public class DosenMain02 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DataDosen02 listDosen = new DataDosen02();
        boolean running = true;

        while (running) {
            System.out.println("\nMenu:");
            System.out.println("1. Tambah Data Dosen");
            System.out.println("2. Tampilkan Data Dosen");
            System.out.println("3. Sorting ASC (Usia Muda ke Tua)");
            System.out.println("4. Sorting DSC (Usia Tua ke Muda)");
            System.out.println("5. Keluar");
            System.out.print("Pilih menu: ");
            int pilihan = sc.nextInt();
            sc.nextLine();

            if (pilihan == 1) {
                System.out.print("Masukkan Kode Dosen: ");
                String kode = sc.nextLine();
                System.out.print("Masukkan Nama Dosen: ");
                String nama = sc.nextLine();
                System.out.print("Masukkan Jenis Kelamin (Laki-laki
= true, Perempuan = false): ");
                boolean jk = sc.nextBoolean();
                System.out.print("Masukkan Usia Dosen: ");
                int usia = sc.nextInt();

                Dosen02 dsn = new Dosen02(kode, nama, jk, usia);
                listDosen.tambah(dsn);
            } else if (pilihan == 2) {
                listDosen.tampil();
            } else if (pilihan == 3) {
                listDosen.SortingASC();
                listDosen.tampil();
            } else if (pilihan == 4) {
                listDosen.sortingDSC();
                listDosen.tampil();
            } else if (pilihan == 5) {
                System.out.println("Program selesai.");
                running = false;
            } else {
                System.out.println("Pilihan tidak valid, silakan
coba lagi.");
            }
        }
    }
}
```

2. Dosen02.java

```
package Tugas;

public class Dosen02 {
    String kode;
    String nama;
    boolean jenisKelamin;
    int usia;

    // Konstruktor
    public Dosen02(String kd, String name, boolean jk, int age)
    {
        this.kode = kd;
        this.nama = name;
        this.jenisKelamin = jk;
        this.usia = age;
    }

    // Menampilkan informasi dosen
    public void tampil() {
        System.out.println("Kode: " + kode);
        System.out.println("Nama: " + nama);
        System.out.println("Jenis Kelamin: " + (jenisKelamin ?
"Laki-laki" : "Perempuan"));
        System.out.println("Usia: " + usia);
        System.out.println("-----");
    }
}
```

3. DataDosen02.java

```

package Tugas;

public class DataDosen02 {
    Dosen02[] dataDosen02 = new Dosen02[10];
    int idx = 0;

    public void tambah(Dosen02 dsn) {
        if (idx < dataDosen02.length) {
            dataDosen02[idx] = dsn;
            idx++;
        } else {
            System.out.println("Data dosen sudah penuh!");
        }
    }

    public void tampil() {
        for (int i = 0; i < idx; i++) {
            dataDosen02[i].tampil();
        }
    }

    public void SortingASC() {}
    public void sortingDSC() {}
    public void insertionSort() {}
}

```

4. Output

```

Menu:
1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting ASC (Usia Muda ke Tua)
4. Sorting DSC (Usia Tua ke Muda)
5. Keluar
Nama: Bagus
Jenis Kelamin: Laki-laki
Usia: 33
-----
Kode: 456
Nama: Wati
Jenis Kelamin: Perempuan
Usia: 34
-----

Menu:
1. Tambah Data Dosen
2. Tampilkan Data Dosen
Nama: Bagus
Jenis Kelamin: Laki-laki
Usia: 33
-----
Kode: 456
Nama: Wati
Jenis Kelamin: Perempuan
Usia: 34
-----

Menu:
1. Tambah Data Dosen
Nama: Bagus
Jenis Kelamin: Laki-laki
Usia: 33
-----
Kode: 456
Nama: Wati
Jenis Kelamin: Perempuan
Usia: 34
-----

Nama: Bagus
Jenis Kelamin: Laki-laki
Usia: 33
Nama: Bagus
Jenis Kelamin: Laki-laki
Nama: Bagus
Jenis Kelamin: Laki-laki
Usia: 33
-----

Kode: 456
Nama: Wati
Jenis Kelamin: Perempuan
Usia: 34
-----

Menu:
1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting ASC (Usia Muda ke Tua)
4. Sorting DSC (Usia Tua ke Muda)
5. Keluar
Pilih menu: 

```

Link Github

<https://github.com/baynobu/ALSD/tree/53a05bc6718c2fdf52494fb9a06460c56f582371/Pertemuan%206>