

LAPORAN PRAKTIKUM

JOBSHEET 6



VANESA MARDIANA PUTRI

244107020129 / 23

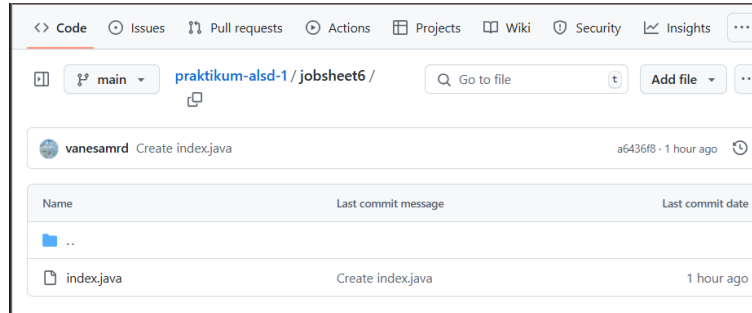
KELAS TI 1B

**PRODI D-IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG**

1. Percobaan 1

Langkah – Langkah :

1. Buat file baru di repository.



2. Buat class Soring(no.absen) dan tambahkan atributnya.

```
public class Sorting23 {  
    int [] data;  
    int jumData;
```

3. Buatlah konstruktor dengan parameter Data[] dan jmlDat

```
Sorting23 (int Data[], int jmlDat){  
    jumData=jmlDat;  
    data=new int [jumData];  
    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=0;j<jumData-i-1;j++){  
            if (data[j-i]>data[j]) {  
                temp=data[j];  
                data[j]=data[j-i];  
                data[j-i]=temp;  
            }  
        }  
    }  
}
```

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

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

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

```
public class SortingMain23 {
    Run | Debug
    public static void main(String[] args) {
        int a[] = {20, 10, 2, 7, 12};
    }
}
```

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

```
Sorting23 dataurut1 = new Sorting23(a, a.length);
```

8. Lakukan pemanggilan method bubbleSort dan tampil.

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

9. Jalankan program.

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

2. Percobaan 2

Langkah – Langkah :

1. 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 parameteranya.

```
Sorting23 dataurut2 = new Sorting23(b, b.length);
```

4. Lakukan pemanggilan method SelectionSort dan tampil.

```

System.out.println(x:"Data awal 2: ");
dataur2.tampil();
dataur2.SelectionSort();
System.out.println(x:"Data sudah diurutkan dengan SELECTION SORT (ASC): ");
dataur2.tampil();

```

5. Run program.

```

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

```

3. Percobaan 3

Langkah – Langkah :

1. 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--;
        }
    }
}

```

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 dataur3 yang merupakan instansiasi dari class Sorting, kemudian isi parameternya.

```

Sorting23 dataur3 = new Sorting23(c, c.length);

```

4. Lakukan pemanggilan method insertionSort dan tampil.

```

System.out.println(x:"Data awal 3: ");
dataur3.tampil();
dataur3.insertionSort();
System.out.println(x:"Data sudah diurutkan dengan INSERT SORT (ASC)");
dataur3.tampil();

```

5. Run program

```

Data awal 3:
40
10
4
9
3

Data sudah diurutkan dengan INSERT SORT (ASC)
40
40
40
40
40

```

Jawaban pertanyaan :

1. Merupakan bagian dari bubble sort yang digunakan untuk mengurutkan elemen dalam array secara ascending.

```

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

```

- 2.
3. Digunakan untuk menggeser elemen yang lebih besar ke kanan agar ada tempat untuk memasukkan elemen yang sedang diproses.
4. Digunakan untuk menggeser elemen ke kanan agar memberikan ruang bagi elemen yang sedang dimasukkan.

4. Praktikum 2

Langkah – Langkah :

1. Buatlah class dengan nama Mahasiswa(no absen).

```
public class mahasiswa223 {
```

2. Tambahkan kode berikut.

```

String nim, nama, kelas;  Declare "nama" and all following declarations on a separate
double ipk;

mahasiswa223() {

}

mahasiswa223(String nm, String name, String kls, double ip) {
    nim = nm;
    nama = name;
    kelas = kls;
    ipk = ip;
}

void tampilkanInformasi(){
    System.out.println("NIM: " + nim);  Replace this use of System.out by a logger.
    System.out.println("Nama: " + nama);  Replace this use of System.out by a logger.
    System.out.println("Kelas: " + kelas);  Replace this use of System.out by a logger.
    System.out.println("IPK: " + ipk);  Replace this use of System.out by a logger.
}
}

```

3. Buat class MahasiswaBerprestasi(no.absen) seperti di bawah ini!

```
public class MahasiswaBerprestasi23 {
    mahasiswa223 [] listMhs = new mahasiswa223 [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 (mahasiswa223 m) {
    if (idx < listMhs.length){
        listMhs[idx]=m;
        idx++;
    } else {
        System.out.println(x:"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!

```
void tampil () {
    for (mahasiswa223 m:listMhs){
        m.tampilkanInformasi();
        System.out.println(x:"-----");
    }
}
```

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

```
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) {
                mahasiswa223 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.

```
Run [Debug]
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);    Remove this unused "sc" local variable.
    MahasiswaBerprestasi123 list = new MahasiswaBerprestasi123();

    mahasiswa223 m1 = new mahasiswa223 (nm:"123", name:"Zidan", kls:"2A", ip:3.2);
    mahasiswa223 m2 = new mahasiswa223 (nm:"124", name:"Ayu", kls:"2A", ip:3.5);
    mahasiswa223 m3 = new mahasiswa223 (nm:"125", name:"Sofi", kls:"2A", ip:3.1);
    mahasiswa223 m4 = new mahasiswa223 (nm:"126", name:"Sita", kls:"2A", ip:3.9);
    mahasiswa223 m5 = new mahasiswa223 (nm:"127", name:"Miki", kls:"2A", ip:3.7);

    list.tambah(m1);
    list.tambah(m2);
    list.tambah(m3);
    list.tambah(m4);
    list.tambah(m5);

    System.out.println(x:"Data Mahasiswa Belum Sorting");    Replace this use of System.out by a logger
    list.tampil();
    System.out.println(x:"Data Mahasiswa Setelah Sorting Berdasarkan IPK (DESC)");    Replace this use
    list.bubbleSort();
    list.tampil();
}
```

8. Run program :

```

C:\Users\user\Documents\Praktikum\Java\Lat_Kas\praktikum_12\src>
Data Mahasiswa Belum Sorting
NIM: 123
Nama: Zidan
Kelas: 2A
IPK: 3.2
-----
NIM: 124
Nama: Ayu
Kelas: 2A
IPK: 3.5
-----
NIM: 125
Nama: Sofi
Kelas: 2A
IPK: 3.1
-----
NIM: 126
Nama: Sita
Kelas: 2A
IPK: 3.9
-----
NIM: 127
Nama: Miki
Kelas: 2A
IPK: 3.7
-----

Data Mahasiswa Setelah Sorting Berdasarkan IPK (DESC)
NIM: 126
Nama: Sita
Kelas: 2A
IPK: 3.9
-----
NIM: 127
Nama: Miki
Kelas: 2A
IPK: 3.7
-----
NIM: 124
Nama: Ayu
Kelas: 2A
IPK: 3.5
-----
NIM: 123
Nama: Zidan
Kelas: 2A
IPK: 3.2
-----
NIM: 125
Nama: Sofi
Kelas: 2A
IPK: 3.1
-----

```

Jawaban pertanyaan :

1. A. agar tidak melebihi batas array, karena membandingkan elemen ini dengan yang lain.
B. untuk menghindari perbandingan dngan elemen yang sudah terurut di akhir array.

C. perulangan I akan berlangsung selama 49 kali dan akan ada 49 tahap bubble sort yang ditempuh untuk mengurutkan 50 elemen.

```
for (int i = 0; i < 5; i++) {
    System.out.print(s:"Masukkan NIM Mahasiswa: ");
    String nim = sc.nextLine();
    System.out.print(s:"Masukkan Nama Mahasiswa: ");
    String nama = sc.nextLine();
    System.out.print(s:"Masukkan Kelas Mahasiswa: ");
    String kelas = sc.nextLine();
    System.out.print(s:"Masukkan IPK Mahasiswa: ");
    double ipk = sc.nextDouble();
    sc.nextLine();

    mahasiswa223 m = new mahasiswa223(nim, nama, kelas, ipk);
    list.tambah(m);
}
```

2.

5. Praktikum 3

Langkah – Langkah :

1. Lihat kembali class MahasiswaBerprestasi, dan tambahkan method selectionSort() di dalamnya!

```
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;
            }
        }
        mahasiswa223 tmp = listMhs[idxMin];
        listMhs[idxMin] = listMhs[i];
        listMhs[i] = tmp;
    }
}
```

2. 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(x:"Data yang sudah terurut menggunakan SELECTION SORT (ASC)");
list.SelectionSort();
list.tampil();
```

3. Run program

```
-----
Data yang sudah terurut menggunakan SELECTION SORT (ASC)
NIM: 125
Nama: Sofi
Kelas: 2A
IPK: 3.1
-----
NIM: 123
Nama: Zidan
Kelas: 2A
IPK: 3.2
-----
NIM: 124
Nama: Ayu
Kelas: 2A
IPK: 3.5
-----
NIM: 127
Nama: Miki
Kelas: 2A
IPK: 3.7
-----
NIM: 126
Nama: Sita
Kelas: 2A
IPK: 3.9
-----
```


6. Praktikum 4

Langkah – Langkah :

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 = 1; i < listMhs.length; i++) {  
        mahasiswa223 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(x: "Data yang sudah terurut menggunakan INSERTION SORT (ASC)");  
list.insertionSort();  
list.tampil();
```

3. Run program :

```
-----  
Data yang sudah terurut menggunakan INSERTION SORT (ASC)  
NIM: 125  
Nama: Sofi  
Kelas: 2A  
IPK: 3.1  
-----  
NIM: 123  
Nama: Zidan  
Kelas: 2A  
IPK: 3.2  
-----  
NIM: 124  
Nama: Ayu  
Kelas: 2A  
IPK: 3.5  
-----  
NIM: 127  
Nama: Miki  
Kelas: 2A  
IPK: 3.7  
-----  
NIM: 126  
Nama: Sita  
Kelas: 2A  
IPK: 3.9  
-----
```

Jawaban pertanyaan :

```

void insertionSort() {
    for (int i = 1; i < listMhs.length; i++) {
        mahasiswa223 temp = listMhs[i];
        int j = i;
        //while (j > 0 && listMhs[j-1].ipk > temp.ipk) { This
        while (j > 0 && listMhs[j-1].ipk < temp.ipk) {
            listMhs[j] = listMhs[j-1];
            j--;
        }
        listMhs[j] = temp;
    }
}

```

1.

7. Tugas

```

Menu:
1. Tambah Data Dosen
2. Tampil Data Dosen
3. Sorting ASC (Usia Muda ke Tua)
4. Sorting DSC (Usia Tua ke Muda)
5. Keluar
Pilih menu: 1
Masukkan Kode Dosen: 345
Masukkan Nama Dosen: Bagas
Masukkan Jenis Kelamin (Laki-laki/Perempuan): Laki-Laki
Masukkan Usia Dosen: 50
Data dosen berhasil ditambahkan!

Menu:
1. Tambah Data Dosen
2. Tampil Data Dosen
3. Sorting ASC (Usia Muda ke Tua)
4. Sorting DSC (Usia Tua ke Muda)
5. Keluar
Pilih menu: 2
Kode: 345, Nama: Bagas, Jenis Kelamin: Laki-laki, Usia: 50

Menu:
1. Tambah Data Dosen
2. Tampil Data Dosen
3. Sorting ASC (Usia Muda ke Tua)
4. Sorting DSC (Usia Tua ke Muda)
5. Keluar
Pilih menu: 3
Data Dosen telah diurutkan berdasarkan usia (Muda ke Tua).

Menu:
1. Tambah Data Dosen
2. Tampil Data Dosen
3. Sorting ASC (Usia Muda ke Tua)
4. Sorting DSC (Usia Tua ke Muda)
5. Keluar
Pilih menu: 4
Data Dosen telah diurutkan berdasarkan usia (Tua ke Muda).

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

```