



JOBSHEET - V

SORTING (BUBBLE SORT DAN SELECTION SORT)

5.1 Tujuan Praktikum

1. Mahasiswa mampu memahami algoritma *sorting* pada jenis *Bubble Sort* dan *Selection Sort*.
2. Mahasiswa mampu membuat dan mendeklarasikan struktur algoritma *Bubble sort* dan *Selection Sort*.
3. Mahasiswa mampu menerapkan dan mengimplementasikan algoritma *Bubble sort* dan *Selection Sort*.

5. 2 Mengimplementasikan Sorting menggunakan Object

Waktu : 50 menit

Pada percobaan ini, kita akan mengimplementasikan penggunaan Sorting yaitu bubble sort dan selection sort yang dibedakan pada instansiasi object.

5.2.1 Langkah Praktikum 1

SORTING – BUBBLE SORT

1. Buatlah project baru pada Netbeans dengan nama **Main**
2. Buat class **Sort**, kemudian deklarasikan variabel berikut ini :

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

3. Buatlah konstruktor dengan parameter Data[] dan jumData

```
public Sort(int Data[], int jmlData){
    }
}
```

4. Isi Konstruktor tersebut dengan kode berikut



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

5. Buatlah method **bubbleSort** bertipe void

6. Deklarasikan isi method **bubbleSort** dengan 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;
            }
        }
    }
}
```

7. Buatlah method **tampilData** bertipe void

```
public void tampilData()
```

8. Deklarasikan isi dari method **tampilData**

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

9. Buat array pada class **Main** dengan nama data kemudian isi array tersebut

```
int a[]={15,10,7,22,5};
```

10. Buatlah objek baru dengan nama **urut** yang merupakan instansiasi dari class **Sort**, kemudian isi parameternya

```
Sort urut=new Sort(a, a.length);
```

11. Lakukan pemanggilan method **bubbleSort** dan **tampilData**

12. Jalankan program, dan amati hasilnya!

5.2.2 Verifikasi Hasil Percobaan

```
Data sebelum urut
15 10 7 22 5
Data sesudah urut Bubble sort (ASC)
5 7 10 15 22
```

SORTING – SELECTION SORT

1. Pada class **Sort** yang sudah dibuat di praktikum sebelumnya tambahkan method **selectionSort** yang mengimplementasikan pengurutan menggunakan algoritma selection sort.

```
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;
    }
}
```

2. Lakukan pemanggilan method selectionSort dan tampilData
3. Jalankan program dan amati hasilnya!

5.2.3 Verifikasi Hasil Percobaan

```
run:
Data sebelum urut
15 10 7 22 5
Data sesudah urut Bubble sort (ASC)
5 7 10 15 22
Data sesudah urut Selection sort (ASC)
5 7 10 15 22
BUILD SUCCESSFUL (total time: 1 second)
```

5.2.4 Pertanyaan!

1. Jelaskan maksud dari kode berikut

```
BubbleSort urut=new BubbleSort(a,a.length);
```

2. Jelaskan fungsi kode program berikut

```
temp=data[j];
data[j]=data[j-1];
data[j-1]=temp;
```

4. Tunjukkan kode program yang merupakan algoritma pencarian nilai minimum pada selection sort!
5. Kerjakan kembali program pengurutan menggunakan algoritma bubble sort dan SelectionSort di atas dengan pengurutan secara DESC!

5.3 Langkah Praktikum 2 (Menggunakan Array of Object)

Waktu : 50 menit

5.3.1 Membuat Class Mahasiswa

1. Perhatikan diagram class Mahasiswa di bawah ini! Diagram class ini yang selanjutnya akan dibuat sebagai acuan dalam membuat kode program class Mahasiswa.

Mahasiswa
nama: String thnMasuk: int umur: int ipk: double
Mahasiswa(n: String, t: int, u: int, i: double) tampil(): void

2. Buat class Mahasiswa seperti di bawah ini!

```

1  package minggu5;
2
3  public class Mahasiswa {
4      String nama;
5      int thnMasuk, umur;
6      double ipk;
7
8      Mahasiswa(String n, int t, int u, double i){
9          nama = n;
10         thnMasuk = t;
11         umur = u;
12         ipk = i;
13     }
14
15     void tampil(){
16         System.out.println("Nama = "+nama);
17         System.out.println("Tahun Masuk = "+thnMasuk);
18         System.out.println("Umur = "+umur);
19         System.out.println("IPK = "+ipk);
20     }
21 }
    
```

5.3.2 Membuat Class DaftarMahasiswaBerprestasi

1. Di dalam class DaftarMahasiswaBerprestasi terdapat daftar mahasiswa-mahasiswa yang dinyatakan sebagai mahasiswa berprestasi. Di dalam class ini juga terdapat method yang digunakan untuk mengurutkan data mahasiswa berdasarkan nilai ipk. Selain itu, juga terdapat method untuk menampilkan

semua data mahasiswa-mahasiswa yang berprestasi tersebut. Selain itu juga terdapat method untuk menambahkan data mahasiswa ke dalam daftar mahasiswa berprestasi. Perhatikan diagram class berikut ini!

DaftarMahasiswaBerprestasi
listMhs: Mahasiswa[5] idx: int
tambah(mhs: Mahasiswa): void tampil(): void bubbleSort(): void

2. Buat class DaftarMahasiswaBerprestasi seperti di bawah ini!

```

1  package minggu5;
2
3  public class DaftarMahasiswaBerprestasi {
4      Mahasiswa listMhs[] = new Mahasiswa[5];
5      int idx;
6
7      //setelah ini tuliskan method tambah()
8
9      //setelah ini tuliskan method tampil()
10
11     //setelah ini tuliskan method bubbleSort()
12 }

```

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

```

7      //setelah ini tuliskan method tambah()
8      void tambah(Mahasiswa m){
9          if(idx<listMhs.length){
10             listMhs[idx] = m;
11             idx++;
12          }else{
13             System.out.println("Data sudah penuh!!");
14          }
15      }

```

4. 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.

```

17     //setelah ini tuliskan method tampil()
18     void tampil(){
19         for(Mahasiswa m : listMhs){
20             m.tampil();
21             System.out.println("-----");
22         }
23     }

```

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

```

25 //setelah ini tuliskan method bubbleSort()
26 void bubbleSort(){
27     for(int i=0; i<listMhs.length-1; i++){
28         for(int j=1; j<listMhs.length-i; j++){
29             if(listMhs[j].ipk > listMhs[j-1].ipk){
30                 //di bawah ini proses swap atau penukaran
31                 Mahasiswa tmp = listMhs[j];
32                 listMhs[j] = listMhs[j-1];
33                 listMhs[j-1] = tmp;
34             }
35         }
36     }
37 }

```

6. Sampai tahap ini class DaftarMahasiswaBerprestasi telah lengkap.

5.3.3 Membuat Class Main

1. Buat class Main dan didalamnya buat method main() seperti di bawah ini!

```

1 package minggu5;
2 import java.util.Scanner;
3
4 public class Main {
5     public static void main(String[] args) {
6
7     }
8 }

```

2. Di dalam method main(), buatlah 2 objek dari class Scanner, sebuah objek DaftarMahasiswaBerprestasi dan deklarasi variabel jumlahMhs senilai 5.!

```

6 Scanner s = new Scanner(System.in);
7 Scanner s1 = new Scanner(System.in);
8 DaftarMahasiswaBerprestasi data = new DaftarMahasiswaBerprestasi();
9 int jumMhs = 5;

```

3. Kemudian lakukan perulangan sebanyak 5 kali menggunakan for, untuk memasukan data nama, umur, tahun masuk dan ipk dari tiap mahasiswa, dan menambahkan objek mahasiswa ke dalam objek daftar mahasiswa berprestasi!

```

11 for(int i=0; i<jumMhs; i++){
12     System.out.print("Nama = ");
13     String nama = s1.nextLine();
14     System.out.print("Thn masuk = ");
15     int thn = s.nextInt();
16     System.out.print("Umur = ");
17     int umur = s.nextInt();
18     System.out.print("IPK = ");
19     double ipk = s.nextDouble();
20
21     Mahasiswa m = new Mahasiswa(nama, thn, umur, ipk);
22     data.tambah(m);
23 }

```

4. Coba tampilkan data-data mahasiswa yang telah masuk dalam daftar mahasiswa berprestasi!



```
System.out.println("-----");
System.out.println("Data mahasiswa sebelum sorting= ");
data.tampil();
```

Coba jalankan dan amati hasilnya. Apakah semua data mahasiswa telah tampil?

5. Panggil method bubbleSort() dan tampilkan kembali hasilnya!

```
27 System.out.println("Data mahasiswa setelah sorting desc berdasar ipk= ");
28 data.bubbleSort();
29 data.tampil();
```

Coba jalankan dan amati kembali hasilnya! Apakah semua data kini telah terurut secara menurun berdasar ipk?

5.3.4 Verifikasi Hasil Percobaan

```
Nama = eka
Tahun Masuk = 2020
Umur = 17
IPK = 3.80
Nama = indah
Tahun Masuk = 2020
Umur = 18
IPK = 3.5
Nama = hasan
Tahun Masuk = 2020
Umur = 19
IPK = 3.25
Nama = riki
Tahun Masuk = 2020
Umur = 18
IPK = 2.9
Nama = yuni
Tahun Masuk = 2020
Umur = 17
IPK = 3.9
-----
Data mahasiswa sebelum sorting=
Nama = eka
Tahun Masuk = 2020
Umur = 17
IPK = 3.8
-----
Nama = indah
Tahun Masuk = 2020
Umur = 18
IPK = 3.5
-----
Nama = hasan
Tahun Masuk = 2020
Umur = 19
IPK = 3.25
-----
Nama = riki
Tahun Masuk = 2020
Umur = 18
IPK = 2.9
-----
Nama = yuni
Tahun Masuk = 2020
Umur = 17
IPK = 3.9
-----
Data mahasiswa setelah sorting desc berdasar ipk=
Nama = yuni
Tahun Masuk = 2020
Umur = 17
IPK = 3.9
-----
Nama = eka
Tahun Masuk = 2020
Umur = 17
IPK = 3.8
-----
Nama = indah
Tahun Masuk = 2020
Umur = 18
IPK = 3.5
-----
Nama = hasan
Tahun Masuk = 2020
Umur = 19
IPK = 3.25
-----
Nama = riki
Tahun Masuk = 2020
Umur = 18
IPK = 2.9
-----
Data mahasiswa setelah sorting desc berdasar ipk=
Nama = yuni
Tahun Masuk = 2020
Umur = 17
IPK = 3.9
-----
Nama = eka
Tahun Masuk = 2020
Umur = 17
IPK = 3.8
-----
Nama = indah
Tahun Masuk = 2020
Umur = 18
IPK = 3.5
-----
Nama = hasan
Tahun Masuk = 2020
Umur = 19
IPK = 3.25
-----
Nama = riki
Tahun Masuk = 2020
Umur = 18
IPK = 2.9
-----
```

5.3.5 Menambahkan Proses Selection Sort di dalam Class DaftarMahasiswaBerprestasi

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

```

39 //setelah ini tuliskan method selectionSort()
40 void selectionSort(){
41     for(int i=0; i<listMhs.length-1; i++){
42         int idxMin = i;
43         for(int j=i+1; j<listMhs.length; j++){
44             if(listMhs[j].ipk < listMhs[idxMin].ipk){
45                 idxMin = j;
46             }
47         }
48         //swap
49         Mahasiswa tmp = listMhs[idxMin];
50         listMhs[idxMin] = listMhs[i];
51         listMhs[i] = tmp;
52     }
53 }
    
```

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

```

30 System.out.println("Data mahasiswa setelah sorting asc berdasar ipk= ");
31 data.selectionSort();
32 data.tampil();
    
```

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

5.3.6 Pertanyaan

- 1) Tunjukkan proses pencarian nilai minimum pada algoritma *Selection Sort* yang telah dipraktikkan pada pertemuan ini (bukti *screen shoot* kode program dan penjelasan)!
- 2) Jelaskan perbedaan proses swap yang terjadi pada *Bubble Sort* dengan *Selection Sort*!
- 3) Di dalam method bubbleSort(), terdapat baris program seperti di bawah ini:

```

29 if(listMhs[j].ipk > listMhs[j-1].ipk){
30     //di bawah ini proses swap atau penukaran
31     Mahasiswa tmp = listMhs[j];
32     listMhs[j] = listMhs[j-1];
33     listMhs[j-1] = tmp;
34 }
35 }
    
```

Jelaskan tujuan dari baris program tersebut

- 4) Perhatikan perulangan di dalam bubbleSort() di bawah ini:

```

27 for(int i=0; i<listMhs.length-1; i++){
28     for(int j=1; j<listMhs.length-i; j++){
    
```

- a. Apakah perbedaan antara kegunaan perulangan i dan perulangan j?
 - b. Mengapa syarat dari perulangan i adalah $i < \text{listMhs.length} - 1$?
 - c. Mengapa syarat dari perulangan j adalah $j < \text{listMhs.length} - i$?
 - d. Jika banyak data di dalam listMhs adalah 50, maka berapakah perulangan i akan berlangsung? Dan ada berapa **Tahap** bubble sort yang ditempuh?
- 5) Di dalam method selection sort, terdapat baris program seperti di bawah ini:



```

42     int idxMin = i;
43     for(int j=i+1; j<listMhs.length; j++){
44         if(listMhs[j].ipk < listMhs[idxMin].ipk){
45             idxMin = j;
46         }
47     }

```

Untuk apakah proses tersebut?

5.4 Tugas

Waktu : 100 menit

1. Disebuah sekolah akan melakukan seleksi anggota paskibraka berdasarkan tinggi badan siswa. Buatlah simulasi pengurutan yang tepat untuk membantu panitia memilih anggota paskibraka dengan pengurutan yang dilakukan adalah mulai dari siswa dengan tinggi badan paling tinggi ke rendah. Berikut ini merupakan ilustrasi class diagramnya. (NB: Semua atribut dibuat inputan).

Paskibraka
nama: String
kelas: String
tinggi: int
Paskibraka(a: String, b: String, c: int)
tampilIdentitas(): void

DaftarAnggotaPaskibraka
listPeserta: Paskibraka[10]
id: int
tambah(m: Paskibraka): void
tampil(): void
Sorting(): void

2. Berdasarkan soal no 1, modifikasilah agar program dapat menampilkan nama, kelas, dan tinggi badan siswa yang diterima sebagai anggota paskibraka dengan ketentuan siswa yang lolos seleksi adalah tinggi badannya minimal 170 cm.
3. Seorang pustakawan hendak mengurutkan buku di dalam almari secara berurutan dari terpendek-tertipis hingga tertinggi-tertebal secara vertikal (berdiri). Buat simulasi pengurutan yang dilakukan oleh pustakawan tersebut dengan mengacu data pada tabel, aturan pengurutan berdasarkan metode program, dan *class diagram* berikut!

Judul Buku	Jumlah Halaman	Ukuran Buku
Algoritma pemrograman	1292	B5
Basis data	400	A4
Dasar Pemrograman	294	B3
Rekayasa Perangkat Lunak	824	B2

Dengan ketentuan sebagai berikut:

- a. Buatlah class **Buku**, **PengurutanBuku**, dan **Main**.
- b. Di dalam class **PengurutanBuku**, buat method pengurutan **ascending** berdasar **UkuranBuku** menggunakan **bubble sort** dan buat method



pengurutan **ascending** berdasar **Jumlah Halaman** menggunakan **selection sort**

- c. Ukuran buku harus melalui konversi tinggi buku yang dimaksud dalam satuan ukuran, contoh A4: 21,0 x 29,7 cm sehingga nilai tinggi buku yang terpakai adalah 29,7.