

**LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR
DATA SORTING (BUBBLE, SELECTION, DAN INSERTION
SORT)**



ABIM MUSTAWA

244107020078

KELAS TI-1B

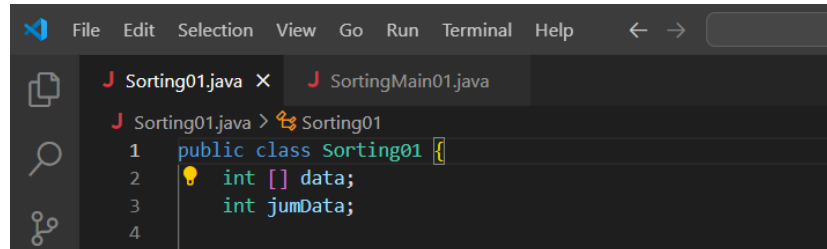
**PRODI D-IV TEKNIK
INFORMATIKA JURUSAN
TEKNOLOGI INFORMASI
POLITEKNIK NEGERI
MALANG 2025**

1. Percobaan Praktikum

1.1 Praktikum 1 – Mengimplementasikan Sorting Menggunakan Object

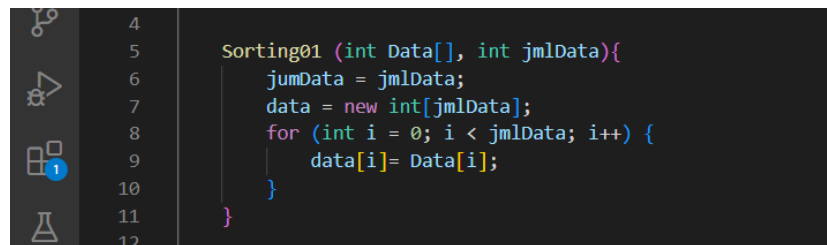
a. Sorting – BUBBLE SORT

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



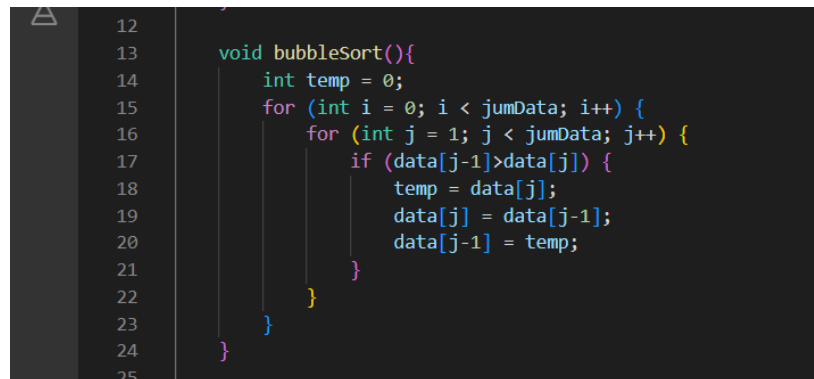
```
File Edit Selection View Go Run Terminal Help
Sorting01.java x SortingMain01.java
Sorting01.java > Sorting01
1 public class Sorting01 {
2     int [] data;
3     int jumData;
4 }
```

3. Buatlah konstruktor dengan parameter Data[] dan jmlDat



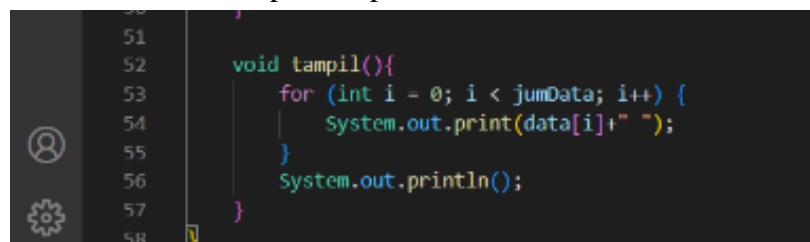
```
4
5     Sorting01 (int Data[], int jmlData){
6         jumData = jmlData;
7         data = new int[jmlData];
8         for (int i = 0; i < jmlData; i++) {
9             data[i] = Data[i];
10        }
11    }
12 }
```

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



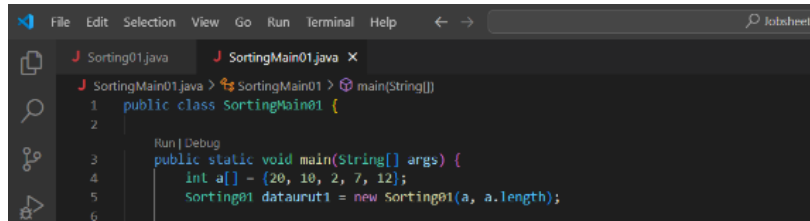
```
12
13     void bubbleSort(){
14         int temp = 0;
15         for (int i = 0; i < jumData; i++) {
16             for (int j = 1; j < jumData; j++) {
17                 if (data[j-1] > data[j]) {
18                     temp = data[j];
19                     data[j] = data[j-1];
20                     data[j-1] = temp;
21                 }
22             }
23         }
24     }
25 }
```

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



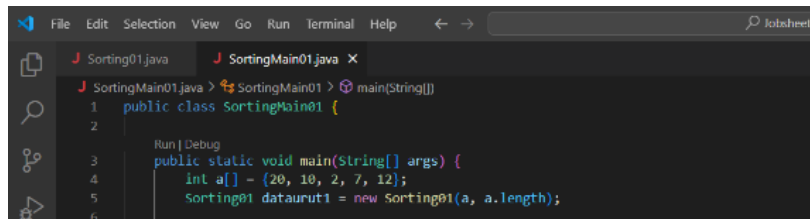
```
51
52     void tampil(){
53         for (int i = 0; i < jumData; i++) {
54             System.out.print(data[i] + " ");
55         }
56         System.out.println();
57     }
58 }
```

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



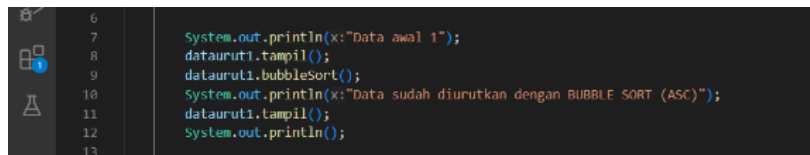
```
1 public class SortingMain01 {
2
3     public static void main(String[] args) {
4         int a[] = {20, 10, 2, 7, 12};
5         Sorting01 dataurut1 = new Sorting01(a, a.length);
6     }
```

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



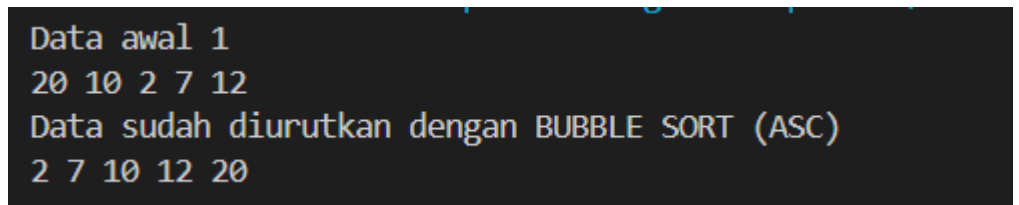
```
1 public class SortingMain01 {
2
3     public static void main(String[] args) {
4         int a[] = {20, 10, 2, 7, 12};
5         Sorting01 dataurut1 = new Sorting01(a, a.length);
6     }
```

8. Lakukan pemanggilan method bubbleSort dan tampil



```
6
7     System.out.println(x:"Data awal 1");
8     dataurut1.tampil();
9     dataurut1.bubbleSort();
10    System.out.println(x:"Data sudah diurutkan dengan BUBBLE SORT (ASC)");
11    dataurut1.tampil();
12    System.out.println();
13 }
```

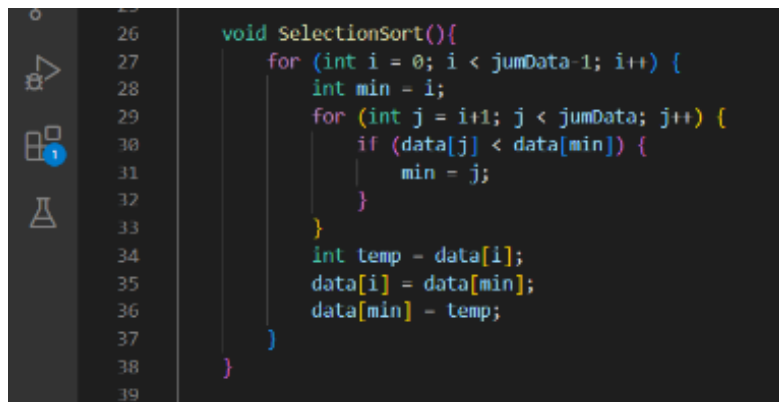
9. Jalankan program, dan amati hasilnya!



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

b. Sorting – SELECTION SORT

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



```
26 void selectionSort(){
27     for (int i = 0; i < jumData-1; i++) {
28         int min = i;
29         for (int j = i+1; j < jumData; j++) {
30             if (data[j] < data[min]) {
31                 min = j;
32             }
33         }
34         int temp = data[i];
35         data[i] = data[min];
36         data[min] = temp;
37     }
38 }
39 }
```

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

```

13
14     int b[] = {30, 20, 2, 8, 14};
15     Sorting01 dataurut2 = new Sorting01(b, b.length);
16
17     System.out.println(x:"Data awal 2");

```

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

```

13
14     int b[] = {30, 20, 2, 8, 14};
15     Sorting01 dataurut2 = new Sorting01(b, b.length);
16
17     System.out.println(x:"Data awal 2");

```

4. Lakukan pemanggilan method SelectionSort dan tampil

```

17     System.out.println(x:"Data awal 2");
18     dataurut2.tampil();
19     dataurut2.SelectionSort();
20     System.out.println(x:"Data sudah diurutkan dengan SELECTION SORT (ASC)");
21     dataurut2.tampil();
22     System.out.println();
23

```

5. Jalankan program dan amati hasilnya!

```

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

```

c. Sorting – INSERTION SORT

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

```

39
40     void insertionSort(){
41         for (int i = 1; i < data.length; i++) {
42             int temp = data[i];
43             int j = i - 1;
44             while (j >= 0 && data[j] > temp) {
45                 data[j+1] = data[j];
46                 j--;
47             }
48             data[j+1] = temp;
49         }
50     }
51

```

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

```

23
24     int c[] = {40, 10, 4, 9, 3};
25     Sorting01 dataurut3 = new Sorting01(c, c.length);
26

```

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

```

23
24     int c[] = {40, 10, 4, 9, 3};
25     Sorting01 dataurut3 = new Sorting01(c, c.length);
26

```

4. Lakukan pemanggilan method insertionSort dan tampil

```

26
27     System.out.println(x:"Data awal 3");
28     dataurut3.tampil();
29     dataurut3.insertionSort();
30     System.out.println(x:"Data sudah diurutkan dengan INSERTION SORT (ASC)");
31     dataurut3.tampil();
32     System.out.println();
33 }
34

```

5. Jalankan program dan amati hasilnya!

```

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

```

Pertanyaan

1. Jelaskan fungsi kode program berikut

Jawab :

```

if (data[j-1]>data[j]) {
    temp = data[j];
    data[j] = data[j-1];
    data[j-1] = temp;
}

```

Kode ini membandingkan dua angka yang bersebelahan. Jika angka di sebelah kiri (data[j-1]) lebih besar dari angka di sebelah kanan (data[j]), maka posisinya akan ditukar.

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

Jawab :

Kode yang menunjukkan algoritma pencarian nilai minimum :

```

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

Jawab :

Kondisi $j \geq 0$ memastikan bahwa tidak keluar dari batas array saat membandingkan elemen. Kondisi $\text{data}[j] > \text{temp}$ berarti selama nilai di kiri lebih besar dari elemen yang sedang diproses, maka elemen tersebut harus digeser ke kanan.

4. Pada Insertion sort, apakah tujuan dari perintah $\text{data}[j + 1] = \text{temp}$;

Jawab :

Kode tersebut digunakan untuk memasukan (disisipkan) elemen ke posisi yang benar.

1.2 Praktikum 2 – Mengurutkan Data Mahasiswa Berdasarkan IPK

Perhatikan diagram class Mahasiswa di bawah ini! Diagram class ini yang selanjutnya akan dibuat sebagai acuan dalam membuat kode program class Mahasiswa. Mahasiswa nim: String nama: String kelas: String ipk: double Mahasiswa() Mahasiswa(nm: String, name: String, kls: String, ip: double) tampilInformasi(): void

Berdasarkan class diagram di atas, kita akan membuat sebuah class Mahasiswa yang berfungsi untuk membuat objek mahasiswa yang akan dimasukkan ke dalam sebuah array. Terdapat sebuah konstruktor default dan berparameter dan juga fungsi tampil() untuk menampilkan semua attribute yang ada.

Selanjutnya class diagram di atas merupakan representasi dari sebuah class yang berfungsi untuk melakukan operasi-operasi dari objek array mahasiswa, misalkan untuk menambahkan objek mahasiswa, menampilkan semua data mahasiswa, dan juga untuk mengurutkan menggunakan Teknik bubble sort berdasarkan nilai IPK mahasiswa.

a. Sorting – BUBBLE SORT

1. Buatlah class dengan nama Mahasiswa.<No Presensi>
2. Untuk lebih jelasnya class tersebut dapat dilihat pada potongan kode di bawah ini

```

Mahasiswa01.java > Mahasiswa01
1  public class Mahasiswa01 {
2      String nim, nama, kelas;
3      double ipk;
4
5      Mahasiswa01(){
6
7      }
8
9      Mahasiswa01(String nm, String name, String kls, double ip){
10         nim = nm;
11         nama = name;
12         kelas = kls;
13         ipk = ip;
14     }
15
16     void tampilInformasi(){
17         System.out.println("Nama : " + nama);
18         System.out.println("NIM : " + nim);
19         System.out.println("Kelas : " + kelas);
20         System.out.println("IPK : " + ipk);
21     }
22
23
24 }
25

```

3. Buat class MahasiswaBerprestasi seperti di bawah ini!
4. Tambahkan method tambah() di dalam class tersebut! Method tambah() digunakan untuk menambahkan objek dari class Mahasiswa ke dalam atribut listMhs
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.
6. Tambahkan method bubbleSort() di dalam class tersebut

```

MahasiswaBerprestasi01.java > MahasiswaBerprestasi01 > bubbleSort()
1  public class MahasiswaBerprestasi01 {
2      Mahasiswa01 [] listMhs= new Mahasiswa01[5];
3      int idx;
4
5      void tambah(Mahasiswa01 m){
6          if (idx < listMhs.length) {
7              listMhs[idx] = m;
8              idx++;
9          }else{
10             System.out.println(x:"Data sudah penuh");
11         }
12     }
13
14     void tampil () {
15         for (Mahasiswa01 m : listMhs) {
16             m.tampilInformasi();
17             System.out.println(x:"-----");
18         }
19     }
20
21     void bubbleSort(){
22         for (int i = 0; i < listMhs.length-1; i++) {
23             for (int j = 1; j < listMhs.length; j++) {
24                 if (listMhs[j].ipk > listMhs[j-1].ipk) {
25                     Mahasiswa01 tmp = listMhs[j];
26                     listMhs[j] = listMhs[j-1];
27                     listMhs[j-1] = tmp;
28                 }
29             }
30         }
31     }
32 }
33

```

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.


```

MahasiswaDemo01.java > MahasiswaDemo01 > main(String[])
1  public class MahasiswaDemo01 {
    Run | Debug
2      public static void main(String[] args) {
3          MahasiswaBerprestasi01 list = new MahasiswaBerprestasi01();
4          Mahasiswa01 m1 = new Mahasiswa01(nm:"123", name:"Zidan", kls:"2A", ip:3.2);
5          Mahasiswa01 m2 = new Mahasiswa01(nm:"124", name:"Ayu", kls:"2A", ip:3.5);
6          Mahasiswa01 m3 = new Mahasiswa01(nm:"125", name:"Sofi", kls:"2A", ip:3.1);
7          Mahasiswa01 m4 = new Mahasiswa01(nm:"126", name:"Sita", kls:"2A", ip:3.9);
8          Mahasiswa01 m5 = new Mahasiswa01(nm:"127", name:"Miki", kls:"2A", ip:3.7);
9
10         list.tambah(m1);
11         list.tambah(m2);
12         list.tambah(m3);
13         list.tambah(m4);
14         list.tambah(m5);
15
16         System.out.println(x:"Data mahasiswa sebelum sorting : ");
17         list.tampil();
18
19         System.out.println(x:"Data mahasiswa setelah sorting berdasarkan ipk (DESC) : ");
20         list.bubbleSort();
21         list.tampil();
22     }
23 }
24

```

<p>Data mahasiswa sebelum sorting :</p> <p>Nama : Zidan NIM : 123 Kelas : 2A IPK : 3.2</p> <p>-----</p> <p>Nama : Ayu NIM : 124 Kelas : 2A IPK : 3.5</p> <p>-----</p> <p>Nama : Sofi NIM : 125 Kelas : 2A IPK : 3.1</p> <p>-----</p> <p>Nama : Sita NIM : 126 Kelas : 2A IPK : 3.9</p> <p>-----</p> <p>Nama : Miki NIM : 127 Kelas : 2A IPK : 3.7</p> <p>-----</p>	<p>Data mahasiswa setelah sorting berdasarkan ipk (DESC) :</p> <p>Nama : Sita NIM : 126 Kelas : 2A IPK : 3.9</p> <p>-----</p> <p>Nama : Miki NIM : 127 Kelas : 2A IPK : 3.7</p> <p>-----</p> <p>Nama : Ayu NIM : 124 Kelas : 2A IPK : 3.5</p> <p>-----</p> <p>Nama : Zidan NIM : 123 Kelas : 2A IPK : 3.2</p> <p>-----</p> <p>Nama : Sofi NIM : 125 Kelas : 2A IPK : 3.1</p> <p>-----</p>
--	---

Pertanyaan

- Perhatikan perulangan di dalam bubbleSort() di bawah ini:
 - Mengapa syarat dari perulangan i adalah $i < \text{listMhs.length} - 1$
 - Mengapa syarat dari perulangan j adalah $j < \text{listMhs.length} - i$
 - Jika banyak data di dalam listMhs adalah 50, maka berapakah perulangan i akan berlangsung? Dan ada berapa Tahap bubble sort yang ditempuh?

Jawab :

- Perulangan i menentukan jumlah tahap yang diperlukan dalam Bubble Sort. Karena dalam setiap tahap, elemen terbesar yang belum terurut akan ditempatkan di posisi yang benar, maka jumlah yang dibutuhkan

- adalah $n-1$, dimana n adalah jumlah elemen dalam array.
- Perulangan j berfungsi untuk melakukan perbandingan antar elemen dan menukar jika diperlukan.
 - Perulangan akan terjadi 1225 kali dan 49 tahap.
2. Modifikasi program diatas dimana data mahasiswa bersifat dinamis (input dari keyboard) yang terdiri dari nim, nama, kelas, dan ipk!

Jawab :

```
J MahasiswaDemo01.java > MahasiswaDemo01 > main(String[])
1  import java.util.Scanner;
2  public class MahasiswaDemo01 {
3      Run | Debug
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6
7          System.out.print(s:"Masukkan jumlah mahasiswa: ");
8          int jumlahMahasiswa = sc.nextInt();
9          sc.nextLine();
10
11          MahasiswaBerprestasi01 list = new MahasiswaBerprestasi01(jumlahMahasiswa);
12
13          for (int i = 0; i < jumlahMahasiswa; i++) {
14              System.out.println("\nMasukkan data mahasiswa ke-" + (i + 1) + ":");
15              System.out.print(s:"NIM: ");
16              String nim = sc.nextLine();
17              System.out.print(s:"Nama: ");
18              String nama = sc.nextLine();
19              System.out.print(s:"Kelas: ");
20              String kelas = sc.nextLine();
21              System.out.print(s:"IPK: ");
22              double ipk = sc.nextDouble();
23              sc.nextLine();
24
25              list.tambah(new Mahasiswa01(nim, nama, kelas, ipk));
26          }
27
28          System.out.println();
29          System.out.println(x:"Data mahasiswa sebelum sorting : ");
30          list.tampil();
31
32          System.out.println(x:"Data mahasiswa setelah sorting berdasarkan ipk (DESC) : ");
33          list.bubbleSort();
34          list.tampil();
35      }
36  }
```

b. Sorting – SELECTION SORT

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

```

void SelectionSort(){
    for (int i = 0; i < listMhs.length; i++) {
        int idxMin=i;
        for (int j = i+1; j < listMhs.length; j++) {
            if (listMhs[j].ipk < listMhs[idxMin].ipk) {
                idxMin=j;
            }
        }
        Mahasiswa01 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 mahasiswa setelah sorting menggunakan SELECTION SORT berdasarkan ipk (DESC) : ");
list.SelectionSort();
list.tampil();

```

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

```

Data mahasiswa sebelum sorting :
Nama : dd
NIM : 33
Kelas : 3e
IPK : 3.4
-----
Nama : ff
NIM : 22
Kelas : 3q
IPK : 3.8
-----
Data mahasiswa setelah sorting menggunakan BUBBLE SORT berdasarkan ipk (DESC) :
Nama : ff
NIM : 22
Kelas : 3q
IPK : 3.8
-----
Nama : dd
NIM : 33
Kelas : 3e
IPK : 3.4
-----
Data mahasiswa setelah sorting menggunakan SELECTION SORT berdasarkan ipk (ASC) :
Nama : dd
NIM : 33
Kelas : 3e
IPK : 3.4
-----
Nama : ff
NIM : 22
Kelas : 3q
IPK : 3.8
-----

```

Pertanyaan

1. Di dalam method selection sort, terdapat baris program seperti di bawah ini:
Untuk apakah proses tersebut, jelaskan!

Jawab :

Kode tersebut digunakan untuk mencari nilai ipk terkecil. Dimulai dengan menganggap elemen pada indeks i adalah yang terkecil ($idxMin = i$). Kemudian, program mengecek elemen-elemen setelahnya. Jika ditemukan elemen dengan ipk yang lebih kecil dari $idxMin=i$, maka $idxMin$ diperbarui ke indeks tersebut. Setelah perulangan selesai, $idxMin$ akan menyimpan posisi elemen dengan ipk terkecil, yang nantinya akan ditukar dengan elemen di posisi i agar tersusun dengan benar.

c. Sorting – INSERTION SORT

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++) {
        Mahasiswa01 temp = listMhs[i];
        int j=i;
        while (j>0 && listMhs[j-1].ipk>temp.ipk) {
            listMhs[j] = listMhs[j-1];
            j--;
        }
        listMhs[j]= temp;
    }
}

```

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

```

System.out.println();
System.out.println(x:"Data mahasiswa setelah sorting menggunakan INSERTION SORT berdasarkan ipk (ASC) : ");
list.InsertionSort();
list.tampil();

```

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

```

Data mahasiswa sebelum sorting :
Nama : qq
NIM : 22
Kelas : 4r
IPK : 3.6
-----
Nama : yy
NIM : 32
Kelas : 4e
IPK : 3.8
-----

Data mahasiswa setelah sorting menggunakan INSERTION SORT
) :
Nama : qq
NIM : 22
Kelas : 4r
IPK : 3.6
-----
Nama : yy
NIM : 32
Kelas : 4e
IPK : 3.8
-----

```

Pertanyaan

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

Jawab :

```

void InsertionSort(){
    for (int i = 1; i < listMhs.length; i++) {
        Mahasiswa01 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. Latihan Praktikum

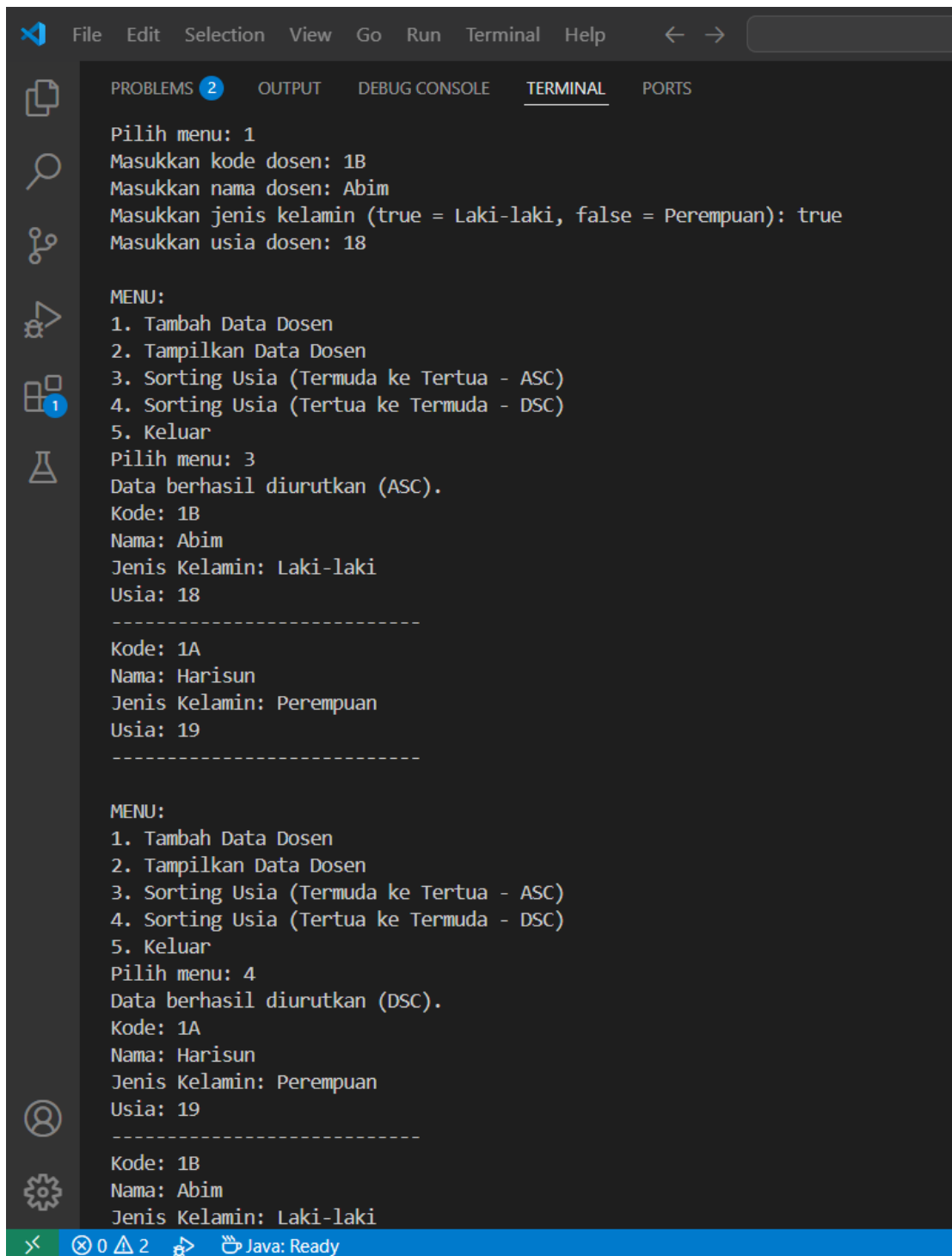
2.1 Latihan 1

Perhatikan class diagram dibawah ini:

- Dosen
 - kode: String
 - nama: String
 - jenisKelamin: Boolean
 - usia: int
 - Dosen(kd: String, name: String, jk: Boolean, age: int)
 - tampil(): void
- DataDosen
 - dataDosen: Dosen[10]
 - idx: int tambah(dsn: Dosen): void
 - tampil(): void
 - SortingASC(): void
 - sortingDSC():void
 - insertionSort():void

Berdasarkan class diagram diatas buatlah menu dikelas main dengan pilihan menu:

1. Tambah data digunakan untuk menambahkan data dosen
2. Tampil data digunakan untuk menampilkan data seluruh dosen
3. Sorting ASC digunakan untuk mengurutkan data dosen berdasarkan usia dimulai dari dosen termuda ke dosen tertua menggunakan bubble Sort.
4. Sorting DSC digunakan untuk mengurutkan data dosen berdasarkan usia dimulai dari tertua ke dosen termuda dapat menggunakan algoritma selection sort atau insertion sort.



```
File Edit Selection View Go Run Terminal Help
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Pilih menu: 1
Masukkan kode dosen: 1B
Masukkan nama dosen: Abim
Masukkan jenis kelamin (true = Laki-laki, false = Perempuan): true
Masukkan usia dosen: 18

MENU:
1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting Usia (Termuda ke Tua - ASC)
4. Sorting Usia (Tua ke Termuda - DSC)
5. Keluar
Pilih menu: 3
Data berhasil diurutkan (ASC).
Kode: 1B
Nama: Abim
Jenis Kelamin: Laki-laki
Usia: 18
-----
Kode: 1A
Nama: Harisun
Jenis Kelamin: Perempuan
Usia: 19
-----

MENU:
1. Tambah Data Dosen
2. Tampilkan Data Dosen
3. Sorting Usia (Termuda ke Tua - ASC)
4. Sorting Usia (Tua ke Termuda - DSC)
5. Keluar
Pilih menu: 4
Data berhasil diurutkan (DSC).
Kode: 1A
Nama: Harisun
Jenis Kelamin: Perempuan
Usia: 19
-----
Kode: 1B
Nama: Abim
Jenis Kelamin: Laki-laki
```

Java: Ready