

Nama : Muhammad Zakaria Haniya

NIM : 244107020135

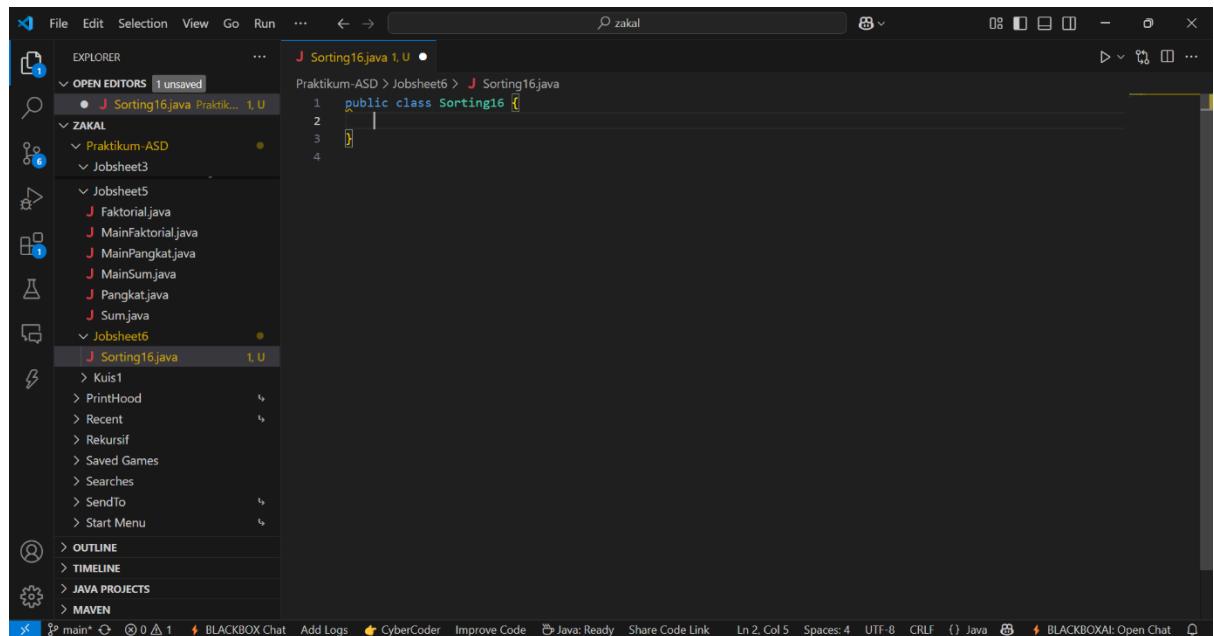
Kelas : TI-1B

Absen : 16

Mengimplementasikan Sorting menggunakan object

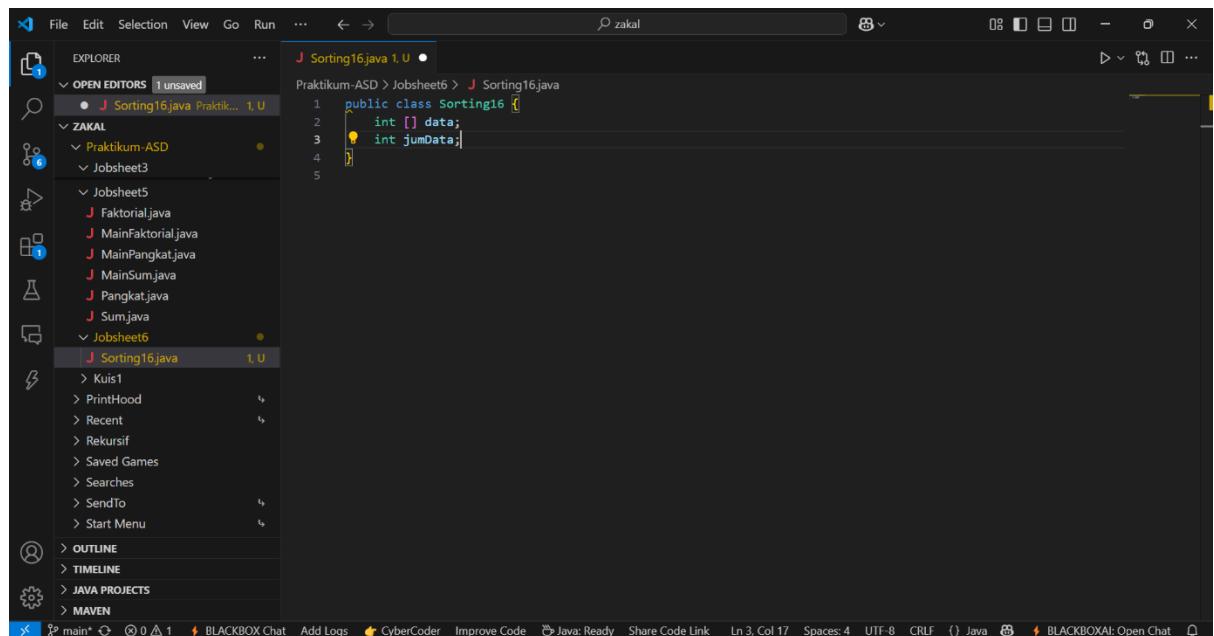
1. SORTING – BUBBLE SORT

- Buat folder baru bernama Jobsheet6 di dalam repository Praktikum ASD



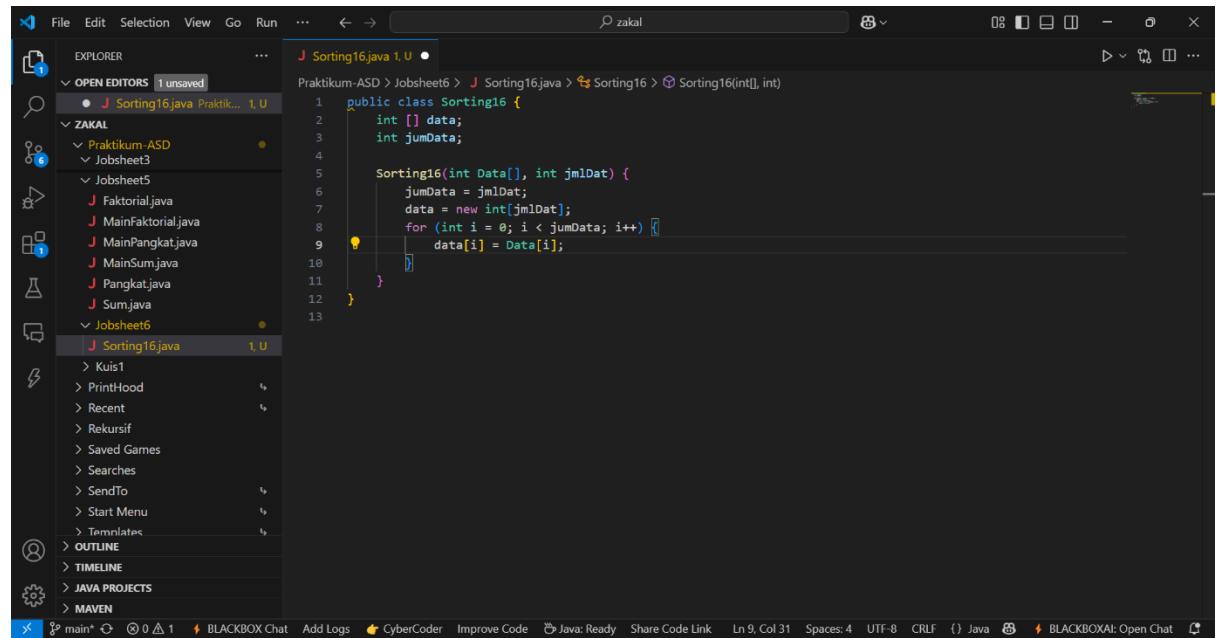
```
public class Sorting16 {
```

- Buat class Sorting, kemudian tambahkan atribut sebagai berikut:



```
public class Sorting16 {
    int [] data;
    int jumlahData;
}
```

- Buatlah konstruktor dengan parameter Data[] dan jmlData



```

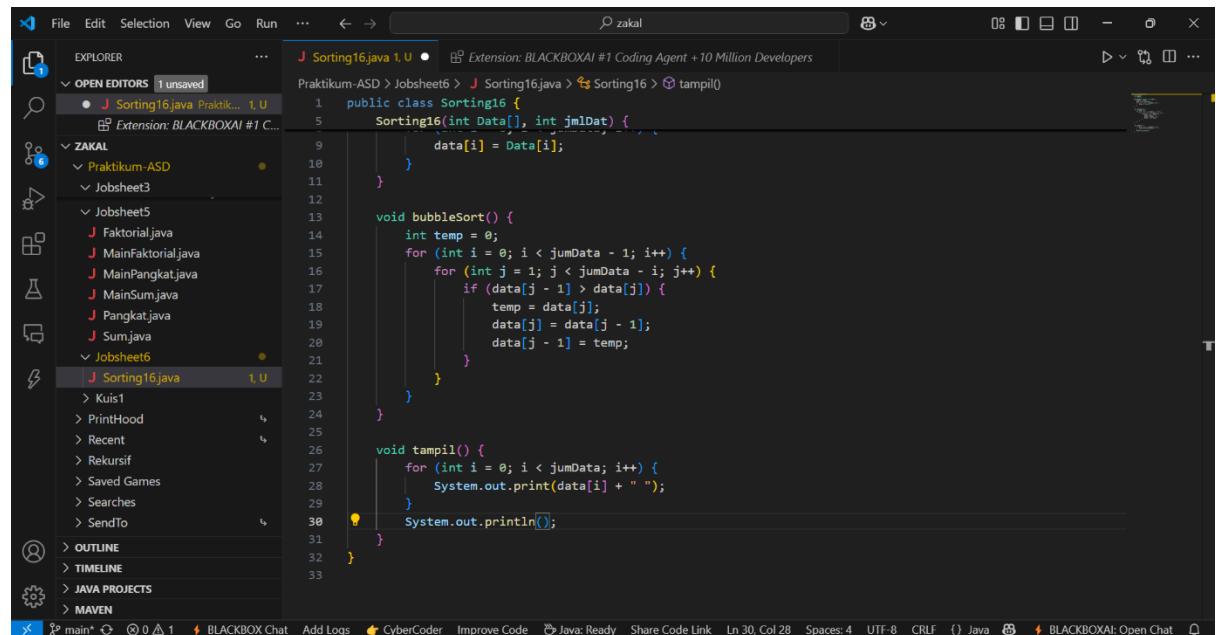
    public class Sorting16 {
        int [] data;
        int jumData;

        Sorting16(int Data[], int jmlData) {
            jumData = jmlData;
            data = new int[jumData];
            for (int i = 0; i < jumData; i++) {
                data[i] = Data[i];
            }
        }
    }

```

The screenshot shows the VS Code interface with the code editor open to the `Sorting16.java` file. The code defines a constructor `Sorting16(int Data[], int jmlData)` that initializes the `data` array and `jumData` variable. The constructor iterates through the input array `Data` and copies its values into the `data` array.

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



```

    public class Sorting16 {
        Sorting16(int Data[], int jmlData) {
            data[i] = Data[i];
        }

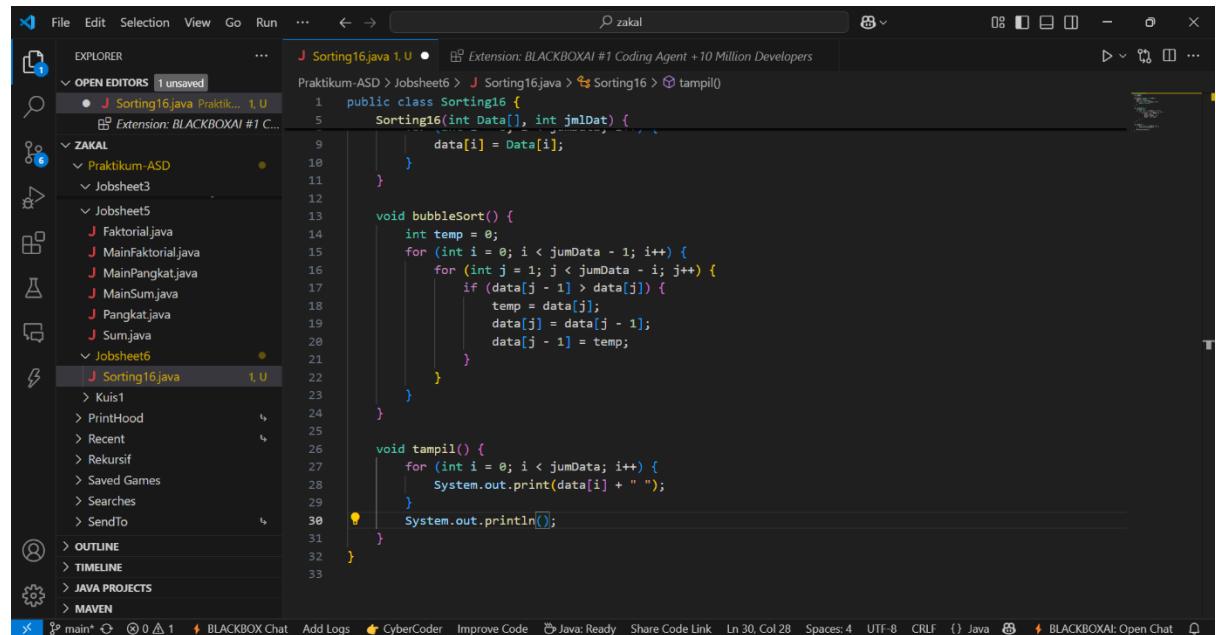
        void bubbleSort() {
            int temp = 0;
            for (int i = 0; i < jumData - 1; i++) {
                for (int j = 1; j < jumData - i; j++) {
                    if (data[j - 1] > data[j]) {
                        temp = data[j];
                        data[j] = data[j - 1];
                        data[j - 1] = temp;
                    }
                }
            }
        }

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

```

The screenshot shows the VS Code interface with the code editor open to the `Sorting16.java` file. It includes a constructor `Sorting16(int Data[], int jmlData)` and two methods: `bubbleSort()` and `tampil()`. The `bubbleSort()` method implements the bubble sort algorithm by comparing adjacent elements and swapping them if they are in the wrong order. The `tampil()` method prints the elements of the `data` array to the console.

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



```

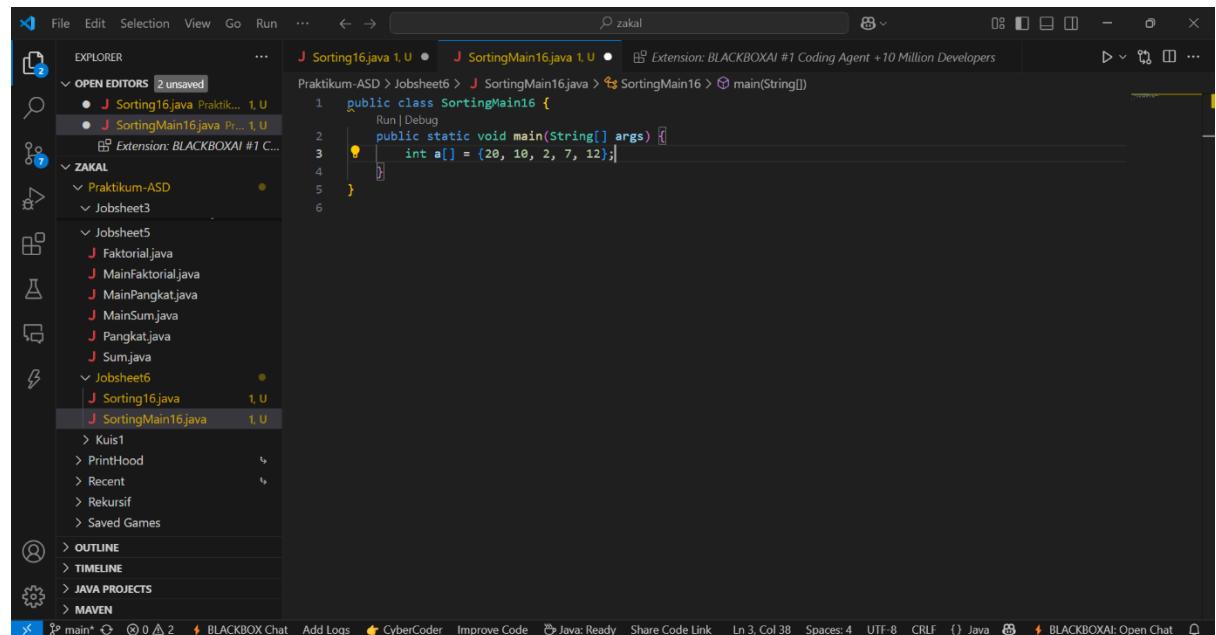
public class Sorting16 {
    Sorting16(int Data[], int jmlData) {
        data[i] = Data[i];
    }
}

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;
            }
        }
    }
}

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

```

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



```

public class SortingMain16 {
    public static void main(String[] args) {
        int a[] = {28, 18, 2, 7, 12};
    }
}

```

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

```

1 public class SortingMain16 {
2     public static void main(String[] args) {
3         int a[] = {28, 18, 2, 7, 12};
4         Sorting16 dataUrut1 = new Sorting16(a, a.length);
5     }
6 }

```

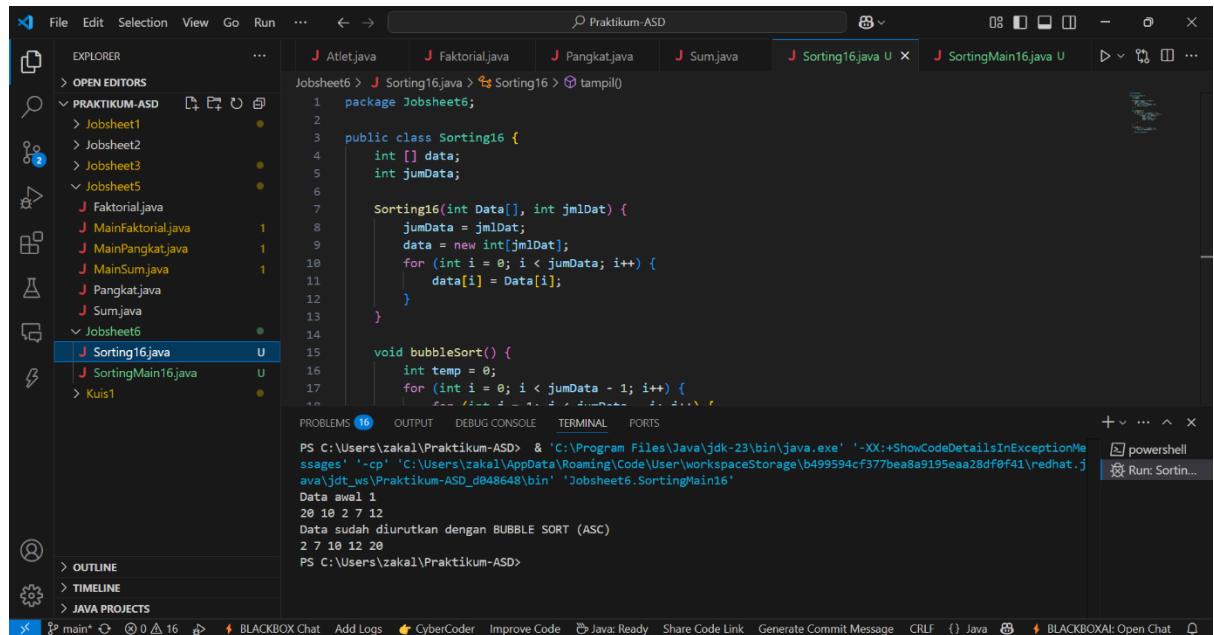
- Lakukan pemanggilan method bubbleSort dan tampil

```

1 public class SortingMain16 {
2     public static void main(String[] args) {
3         int a[] = {28, 18, 2, 7, 12};
4         Sorting16 dataUrut1 = new Sorting16(a, a.length);
5
5         System.out.println("Data awal 1");
6         dataUrut1.tampil();
7         dataUrut1.bubbleSort();
8         System.out.println("Data sudah diurutkan dengan BUBBLE SORT (ASC)");
9         dataUrut1.tampil();
10    }
11 }
12

```

- Jalankan program, dan amati hasilnya!



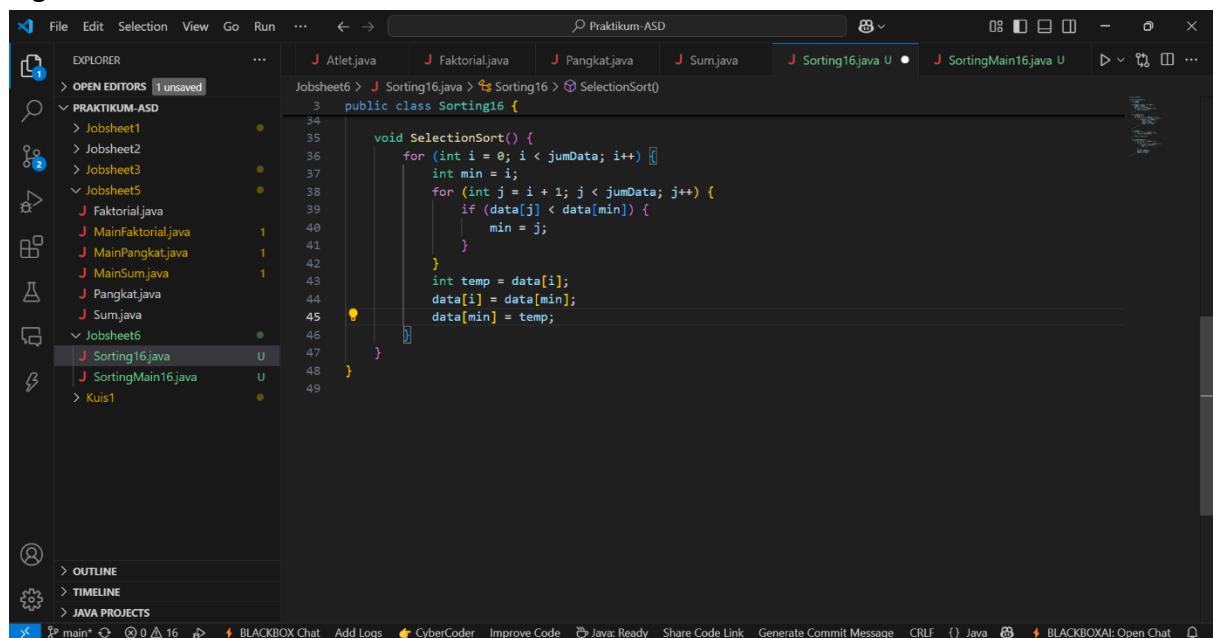
```

File Edit Selection View Go Run ... ← → 🔍 Praktikum-ASD
EXPLORER OPEN EDITORS
PRAKTIKUM-ASD
  > Jobsheet1
  > Jobsheet2
  > Jobsheet3
  > Jobsheet5
    J Faktorial.java
    J MainFaktorial.java 1
    J MainPangkat.java 1
    J MainSum.java 1
    J Pangkat.java
    J Sum.java
  > Jobsheet6
    J Sorting16.java U
    J SortingMain16.java U
  > Kuis1
OUTLINE TIMELINE JAVA PROJECTS
PROBLEMS 16 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\zakal\Praktikum-ASD & 'C:\Program Files\Java\jdk-23\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\zakal\AppData\Roaming\Code\User\workspaceStorage\b499594cf377bea8a9195eaa28df0f41\redhat.ja
va\jdt_ws\Praktikum-ASD_d048648\bin' 'Jobsheet6.SortingMain16'
Data awal 1
20 10 2 7 12
Data sudah diurutkan dengan BUBBLE SORT (ASC)
2 7 10 12 20
PS C:\Users\zakal\Praktikum-ASD

```

2. SORTING – SELECTION SORT

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



```

File Edit Selection View Go Run ... ← → 🔍 Praktikum-ASD
EXPLORER OPEN EDITORS 1 unsaved
PRAKTIKUM-ASD
  > Jobsheet1
  > Jobsheet2
  > Jobsheet3
  > Jobsheet5
    J Faktorial.java
    J MainFaktorial.java 1
    J MainPangkat.java 1
    J MainSum.java 1
    J Pangkat.java
    J Sum.java
  > Jobsheet6
    J Sorting16.java U
    J SortingMain16.java U
  > Kuis1
OUTLINE TIMELINE JAVA PROJECTS
PROBLEMS 16 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\zakal\Praktikum-ASD & 'C:\Program Files\Java\jdk-23\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\zakal\AppData\Roaming\Code\User\workspaceStorage\b499594cf377bea8a9195eaa28df0f41\redhat.ja
va\jdt_ws\Praktikum-ASD_d048648\bin' 'Jobsheet6.Sorting16'
public class Sorting16 {
  ...
  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;
    }
  }
}

```

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

```

File Edit Selection View Go Run ... ← → 🔍 Praktikum-ASD
EXPLORER OPEN EDITORS 2 unsaved
PRAKTIKUM-ASD
> Jobsheet1
> Jobsheet2
> Jobsheet3
> Jobsheet5
J Faktorial.java
J MainFaktorial.java 1
J MainPangkat.java 1
J MainSum.java 1
> Jobsheet6
J Sorting16.java U
J SortingMain16.java 1, U
> Kuis1
Jobsheet6 > J SortingMain16.java > SortingMain16 > main(String[])
1 package Jobsheet6;
2
3 public class SortingMain16 {
4     Run | Debug
5     public static void main(String[] args) {
6         int a [] = {20, 10, 2, 7, 12};
7
8         Sorting16 dataurut1 = new Sorting16(a, a.length);
9
10        System.out.println("Data awal 1");
11        dataurut1.tampil();
12        dataurut1.bubbleSort();
13        System.out.println("Data sudah diurutkan dengan BUBBLE SORT (ASC)");
14        dataurut1.tampil();
15
16        int b [] = {30, 20, 2, 8, 14};
17    }
18
19
20
@ OUTLINE
@ TIMELINE
@ JAVA PROJECTS
main* ⌂ 0 ▲ 17 🔍 BLACKBOX Chat Add Logs CyberCoder Improve Code Java: Ready Share Code Link Generate Commit Message CRLF {} Java 🔍 BLACKBOXAI: Open Chat

```

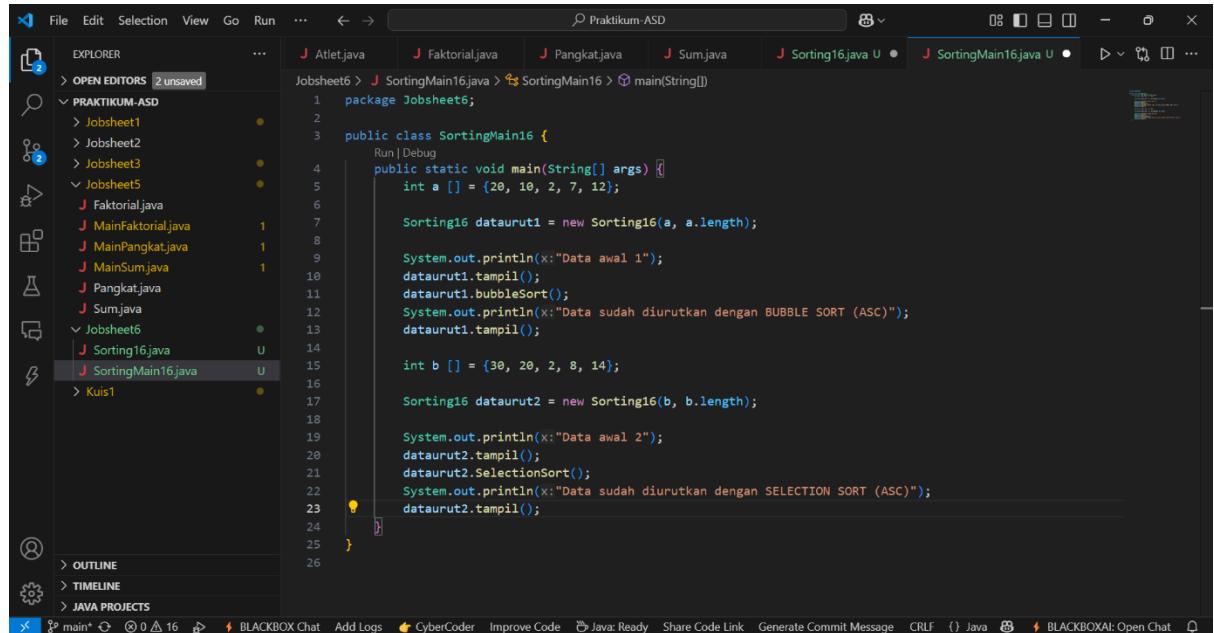
- Buatlah objek baru dengan nama dataurut2 yang merupakan instansiasi dari class Sorting, kemudian isi parameternya

```

File Edit Selection View Go Run ... ← → 🔍 Praktikum-ASD
EXPLORER OPEN EDITORS 2 unsaved
PRAKTIKUM-ASD
> Jobsheet1
> Jobsheet2
> Jobsheet3
> Jobsheet5
J Faktorial.java
J MainFaktorial.java 1
J MainPangkat.java 1
J MainSum.java 1
> Jobsheet6
J Sorting16.java U
J SortingMain16.java 1, U
> Kuis1
Jobsheet6 > J SortingMain16.java > SortingMain16 > main(String[])
1 package Jobsheet6;
2
3 public class SortingMain16 {
4     Run | Debug
5     public static void main(String[] args) {
6         int a [] = {20, 10, 2, 7, 12};
7
8         Sorting16 dataurut1 = new Sorting16(a, a.length);
9
10        System.out.println("Data awal 1");
11        dataurut1.tampil();
12        dataurut1.bubbleSort();
13        System.out.println("Data sudah diurutkan dengan BUBBLE SORT (ASC)");
14        dataurut1.tampil();
15
16        int b [] = {30, 20, 2, 8, 14};
17
18        Sorting16 dataurut2 = new Sorting16(b, b.length);
19    }
20
21
22
@ OUTLINE
@ TIMELINE
@ JAVA PROJECTS
main* ⌂ 0 ▲ 17 🔍 BLACKBOX Chat Add Logs CyberCoder Improve Code Java: Ready Share Code Link Generate Commit Message CRLF {} Java 🔍 BLACKBOXAI: Open Chat

```

- Lakukan pemanggilan method SelectionSort dan tampil



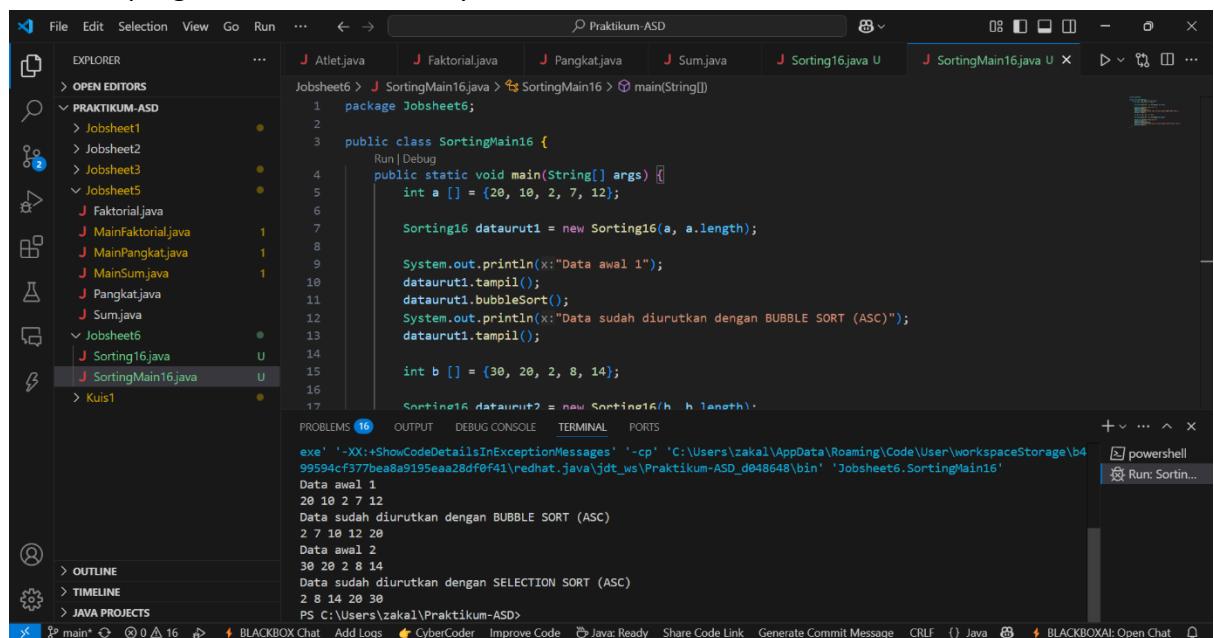
```

File Edit Selection View Go Run ...
OPEN EDITORS 2 unsaved
PRAKTIKUM-ASD
Jobsheet1
Jobsheet2
Jobsheet3
Jobsheet5
Faktorial.java
MainFaktorial.java
MainPangkat.java
MainSum.java
Pangkat.java
Sum.java
Jobsheet6
Sorting16.java
SortingMain16.java
Kuis1

Jobsheet6 > J SortingMain16.java > J main(String[])
1 package Jobsheet6;
2
3 public class SortingMain16 {
4     Run|Debug
5     public static void main(String[] args) {
6         int a [] = {20, 10, 2, 7, 12};
7
8         Sorting16 dataurut1 = new Sorting16(a, a.length);
9
10        System.out.println(x:"Data awal 1");
11        dataurut1.tampil();
12        dataurut1.bubbleSort();
13        System.out.println(x:"Data sudah diurutkan dengan BUBBLE SORT (ASC)");
14        dataurut1.tampil();
15
16        int b [] = {30, 20, 2, 8, 14};
17
18        Sorting16 dataurut2 = new Sorting16(b, b.length);
19
20        System.out.println(x:"Data awal 2");
21        dataurut2.tampil();
22        dataurut2.SelectionSort();
23        System.out.println(x:"Data sudah diurutkan dengan SELECTION SORT (ASC)");
24        dataurut2.tampil();
25    }
26

```

- Jalankan program dan amati hasilnya!



```

File Edit Selection View Go Run ...
OPEN EDITORS
PRAKTIKUM-ASD
Jobsheet1
Jobsheet2
Jobsheet3
Jobsheet5
Faktorial.java
MainFaktorial.java
MainPangkat.java
MainSum.java
Pangkat.java
Sum.java
Jobsheet6
Sorting16.java
SortingMain16.java
Kuis1

Jobsheet6 > J SortingMain16.java > J main(String[])
1 package Jobsheet6;
2
3 public class SortingMain16 {
4     Run|Debug
5     public static void main(String[] args) {
6         int a [] = {20, 10, 2, 7, 12};
7
8         Sorting16 dataurut1 = new Sorting16(a, a.length);
9
10        System.out.println(x:"Data awal 1");
11        dataurut1.tampil();
12        dataurut1.bubbleSort();
13        System.out.println(x:"Data sudah diurutkan dengan BUBBLE SORT (ASC)");
14        dataurut1.tampil();
15
16        int b [] = {30, 20, 2, 8, 14};
17
18        Sorting16 dataurut2 = new Sorting16(b, b.length);
19
20        System.out.println(x:"Data awal 2");
21        dataurut2.tampil();
22        dataurut2.SelectionSort();
23        System.out.println(x:"Data sudah diurutkan dengan SELECTION SORT (ASC)");
24        dataurut2.tampil();
25    }
26

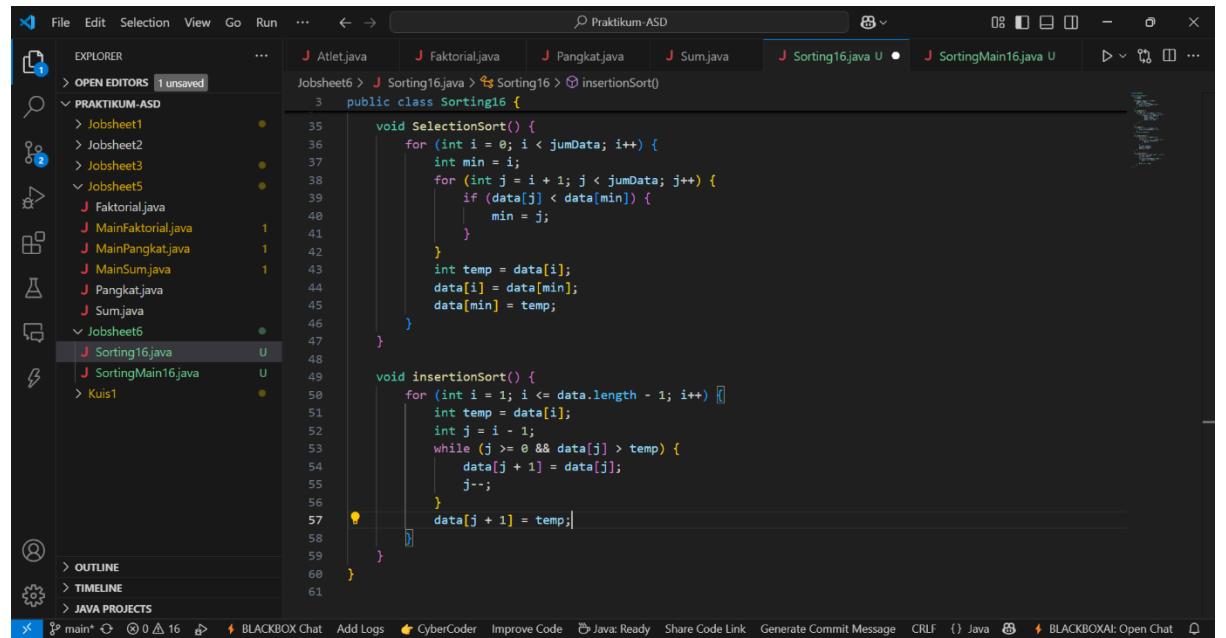
PROBLEMS 16 OUTPUT DEBUG CONSOLE TERMINAL PORTS
exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\zakal\AppData\Roaming\Code\User\workspaceStorage\b49594cf377be8a9195ea28df0f41\redhat.java\jdt_ws\Praktikum-ASD_d048648\bin' 'Jobsheet6.SortingMain16'
Data awal 1
20 10 2 7 12
Data sudah diurutkan 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 C:\Users\zakal\Praktikum-ASD>

```

3. SORTING – INSERTION SORT

- Pada class Sorting yang sudah dibuat di praktikum sebelumnya tambahkan method insertionSort yang mengimplementasikan pengurutan menggunakan

algoritma insertion sort.

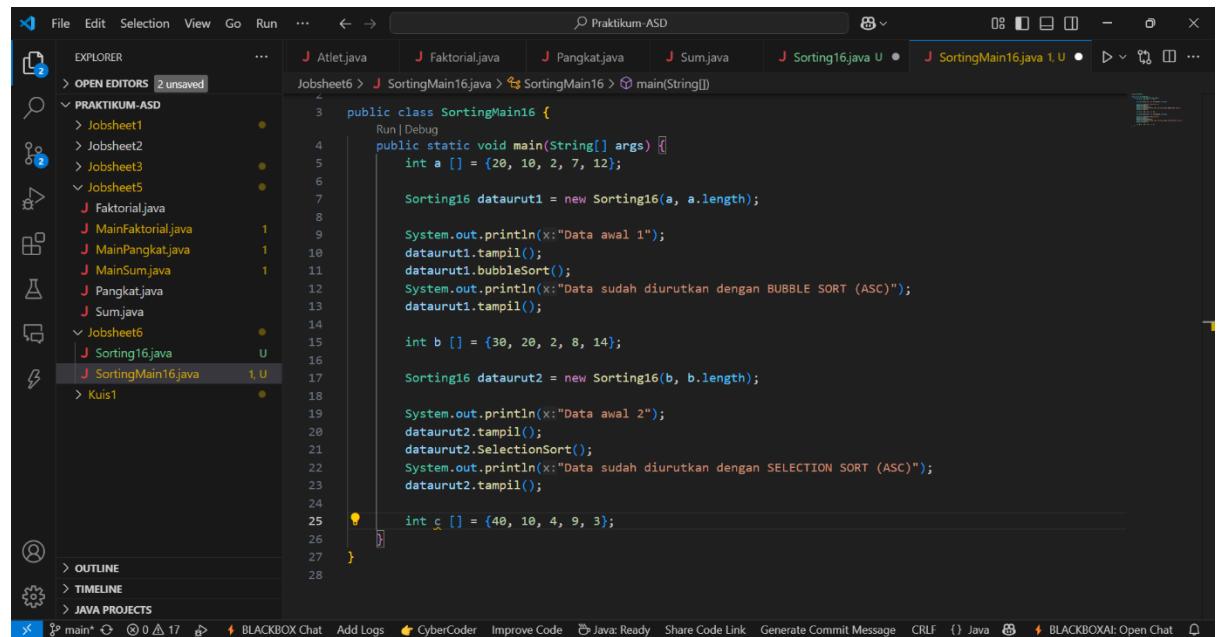


The screenshot shows the Java code for the `SelectionSort` algorithm. The code consists of two main methods: `SelectionSort()` and `insertionSort()`. The `SelectionSort()` method iterates through the array, finding the minimum value in the unsorted portion and swapping it with the current element. The `insertionSort()` method iterates through the array, shifting elements to the right and inserting the current element at the correct position. The code is written in Java and uses standard programming constructs like loops and conditionals.

```
public class Selection16 {
    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;
        }
    }

    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;
        }
    }
}
```

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



The screenshot shows the Java code for the `main` method of the `SortingMain16` class. The code initializes arrays `a`, `b`, and `c` with specific values. It then creates instances of the `Sorting16` class for both arrays and performs bubble sort on `a` and selection sort on `b`. Finally, it prints the sorted arrays `a` and `b` to the console. The code demonstrates how to call the sorting methods from the main program.

```
public class SortingMain16 {
    public static void main(String[] args) {
        int a [] = {20, 10, 2, 7, 12};

        Sorting16 dataurut1 = new Sorting16(a, a.length);

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

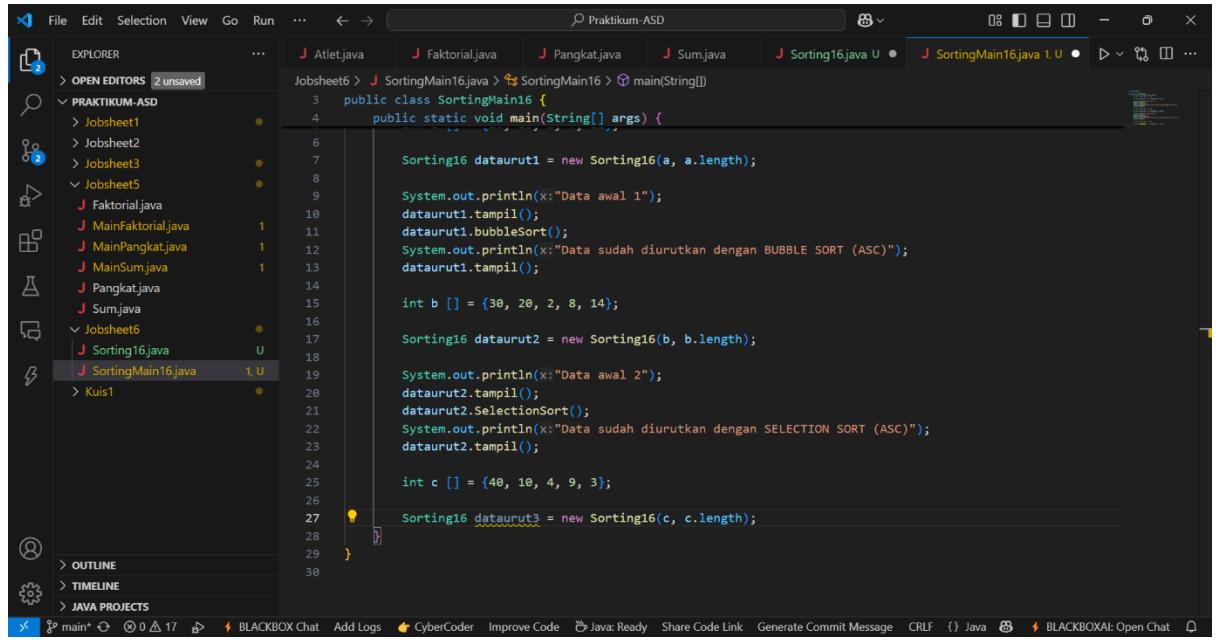
        int b [] = {30, 20, 2, 8, 14};

        Sorting16 dataurut2 = new Sorting16(b, b.length);

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

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

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



```

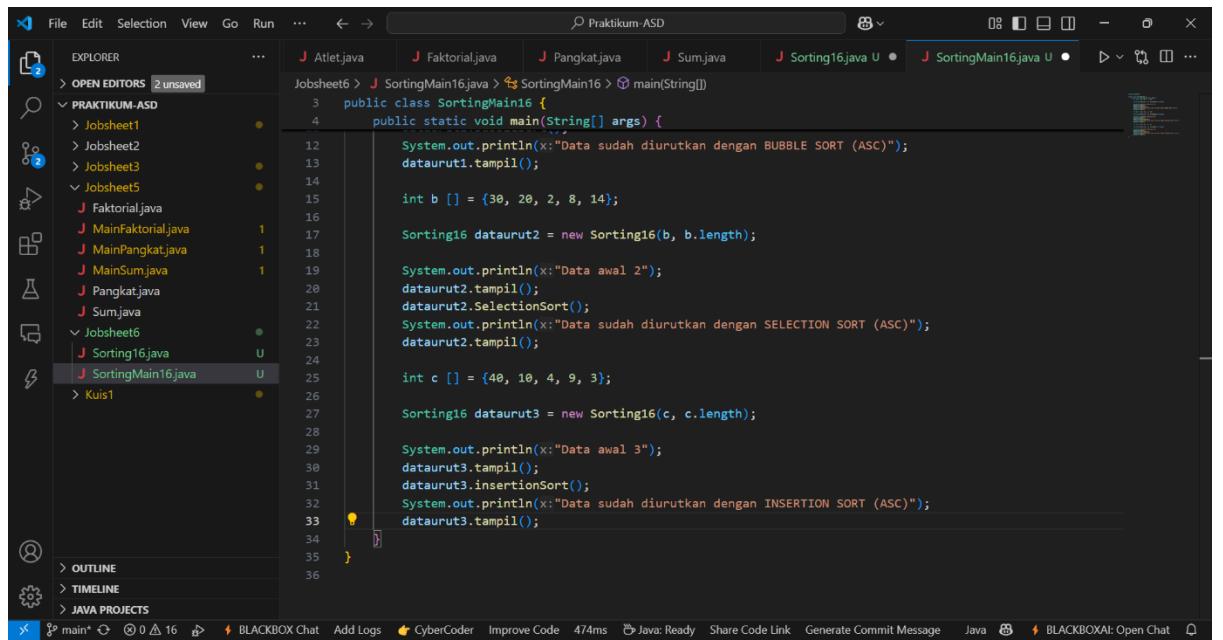
File Edit Selection View Go Run ...
OPEN EDITORS 2 unsaved
PRAKTIKUM-ASD
> Jobsheet1
> Jobsheet2
> Jobsheet3
> Jobsheet5
> Jobsheet6
> J Atlet.java
> J Faktorial.java
> J MainFaktorial.java
> J MainPangkat.java
> J MainSum.java
> J Pangkat.java
> J Sum.java
> J Sorting16.java
> J SortingMain16.java 1, U
> Kuis1
...
public class SortingMain16 {
    public static void main(String[] args) {
        int a [] = {30, 20, 2, 8, 14};
        Sorting16 dataurut1 = new Sorting16(a, a.length);
        System.out.println(x:"Data awal 1");
        dataurut1.tampil();
        dataurut1.bubbleSort();
        System.out.println(x:"Data sudah diurutkan dengan BUBBLE SORT (ASC)");
        dataurut1.tampil();

        int b [] = {30, 20, 2, 8, 14};
        Sorting16 dataurut2 = new Sorting16(b, b.length);
        System.out.println(x:"Data awal 2");
        dataurut2.tampil();
        dataurut2.SelectionSort();
        System.out.println(x:"Data sudah diurutkan dengan SELECTION SORT (ASC)");
        dataurut2.tampil();

        int c [] = {40, 10, 4, 9, 3};
        Sorting16 dataurut3 = new Sorting16(c, c.length);
        System.out.println(x:"Data awal 3");
        dataurut3.tampil();
        dataurut3.insertionSort();
        System.out.println(x:"Data sudah diurutkan dengan INSERTION SORT (ASC)");
        dataurut3.tampil();
    }
}

```

- Lakukan pemanggilan method insertionSort dan tampil



```

File Edit Selection View Go Run ...
OPEN EDITORS 2 unsaved
PRAKTIKUM-ASD
> Jobsheet1
> Jobsheet2
> Jobsheet3
> Jobsheet5
> J Faktorial.java
> J MainFaktorial.java
> J MainPangkat.java
> J MainSum.java
> J Pangkat.java
> J Sum.java
> J Sorting16.java
> J SortingMain16.java 1, U
> Kuis1
...
public class SortingMain16 {
    public static void main(String[] args) {
        int a [] = {30, 20, 2, 8, 14};
        Sorting16 dataurut1 = new Sorting16(a, a.length);
        System.out.println(x:"Data sudah diurutkan dengan BUBBLE SORT (ASC)");
        dataurut1.tampil();

        int b [] = {30, 20, 2, 8, 14};
        Sorting16 dataurut2 = new Sorting16(b, b.length);
        System.out.println(x:"Data awal 2");
        dataurut2.tampil();
        dataurut2.SelectionSort();
        System.out.println(x:"Data sudah diurutkan dengan SELECTION SORT (ASC)");
        dataurut2.tampil();

        int c [] = {40, 10, 4, 9, 3};
        Sorting16 dataurut3 = new Sorting16(c, c.length);
        System.out.println(x:"Data awal 3");
        dataurut3.tampil();
        dataurut3.insertionSort();
        System.out.println(x:"Data sudah diurutkan dengan INSERTION SORT (ASC)");
        dataurut3.tampil();
    }
}

```

- Jalankan program dan amati hasilnya!

```

File Edit Selection View Go Run ... ← → 🔍 Praktikum ASD
EXPLORER J Atlet.java J Faktorial.java J Pangkat.java J Sum.java J Sorting16.java U J SortingMain16.java U ...
OPEN EDITORS PRAKTIKUM-ASD Jobsheet1
Jobsheet2
Jobsheet3
Jobsheet5
J Faktorial.java
J MainFaktorial.java 1
J MainPangkat.java 1
J MainSum.java 1
J Pangkat.java
J Sum.java
Jobsheet6
J Sorting16.java U
J SortingMain16.java U ...
Kuis1
PROBLEMS 16 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Data sudah diurutkan dengan BUBBLE SORT (ASC)
2 7 10 12 20
Data awal
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 C:\Users\zakal\Praktikum-ASD>

```

Pertanyaan :

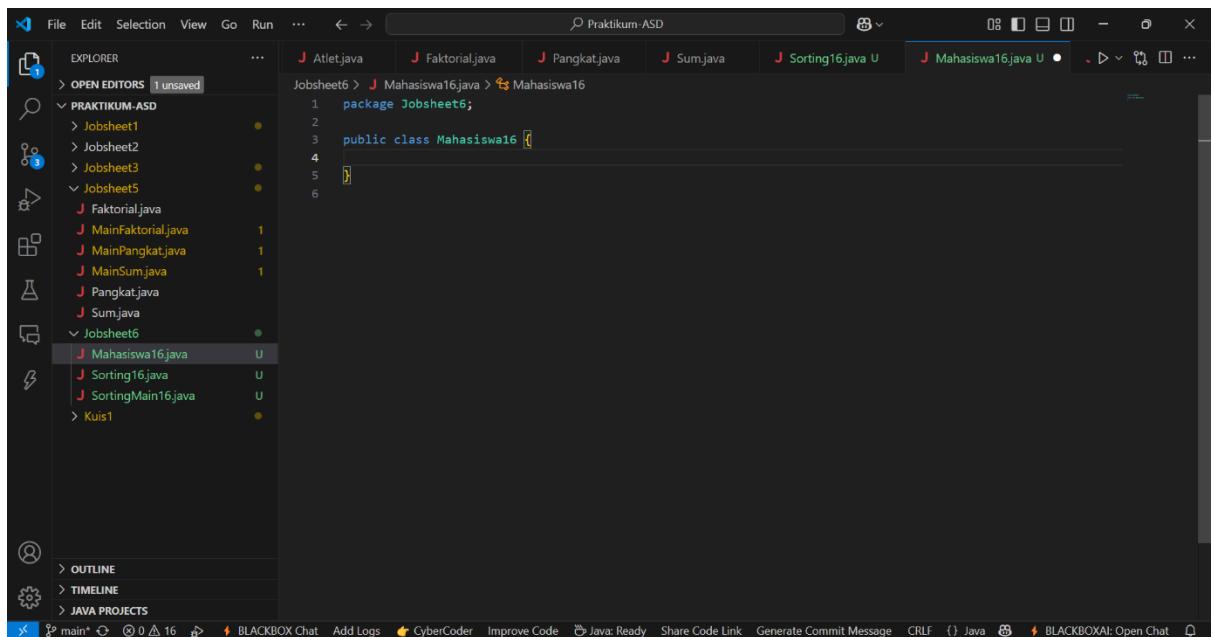
1. Jelaskan fungsi kode program berikut
 - Fungsi dari kode program tersebut adalah untuk menukar jika elemen di kiri lebih besar dari pada yang di kanan
2. Tunjukkan kode program yang merupakan algoritma pencarian nilai minimum pada selection sort!


```

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
 - Mengecek semua elemen di kiri dari temp
 - Menggeser elemen yang lebih besar ke kanan
 - Membuka ruang untuk temp agar bisa disisipkan di posisi yang benar
4. Pada Insertion sort, apakah tujuan dari perintah
 - Tujuan dari perintah tersebut adalah untuk memindahkan elemen ke kanan

Sorting Menggunakan Array of Object

1. Buatlah class dengan nama Mahasiswa

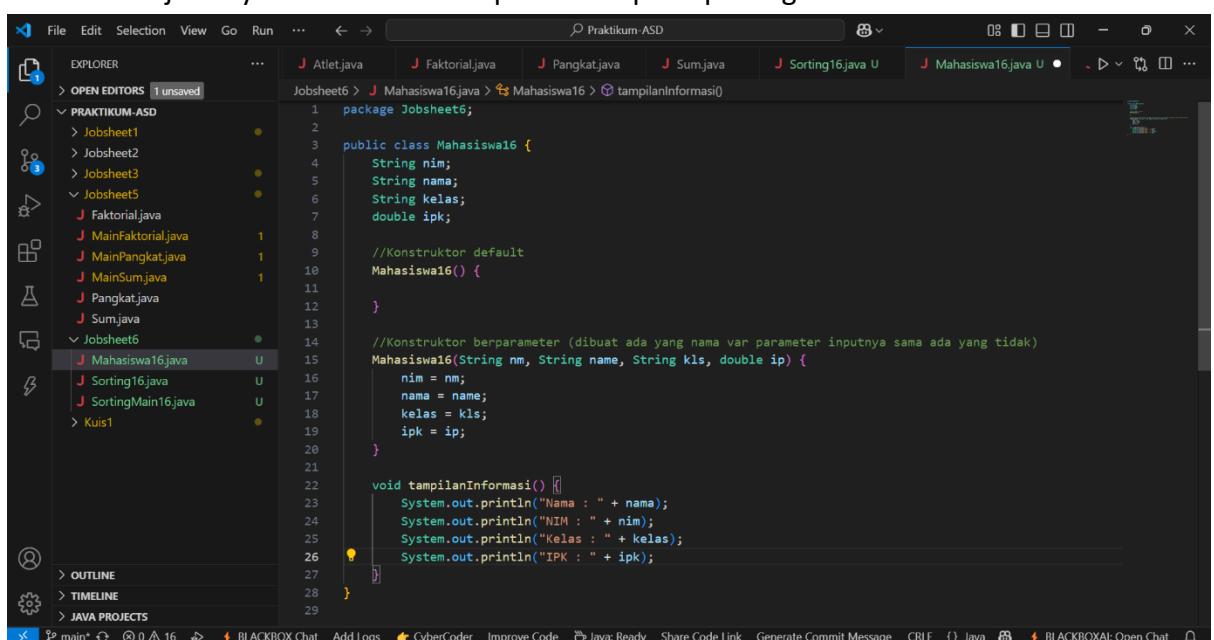


The screenshot shows a Java development environment with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** Praktikum-ASD
- Toolbar:** Standard icons for file operations.
- Left Sidebar (EXPLORER):** Shows the project structure under PRAKTIKUM-ASD, including Jobsheet1, Jobsheet2, Jobsheet3, Jobsheet5, and Jobsheet6. Under Jobsheet6, files like Atlet.java, Faktorial.java, Pangkat.java, Sum.java, Sorting16.java, and Mahasiswa16.java are listed. Mahasiswa16.java is currently selected.
- Right Editor Area:** Displays the code for Mahasiswa16.java:

```
1 package Jobsheet6;
2
3 public class Mahasiswa16 {
4
5 }
```
- Bottom Status Bar:** main*, 0 16, BLACKBOX Chat, Add Logs, CyberCoder, Improve Code, Java: Ready, Share Code Link, Generate Commit Message, CRLF, Java, BLACKBOXAI: Open Chat.

2. Untuk lebih jelasnya class tersebut dapat dilihat pada potongan kode di bawah ini



The screenshot shows the same Java development environment with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** Praktikum-ASD
- Toolbar:** Standard icons for file operations.
- Left Sidebar (EXPLORER):** Shows the project structure under PRAKTIKUM-ASD, including Jobsheet1, Jobsheet2, Jobsheet3, Jobsheet5, and Jobsheet6. Under Jobsheet6, files like Atlet.java, Faktorial.java, Pangkat.java, Sum.java, Sorting16.java, and Mahasiswa16.java are listed. Mahasiswa16.java is currently selected.
- Right Editor Area:** Displays the complete code for Mahasiswa16.java:

```
1 package Jobsheet6;
2
3 public class Mahasiswa16 {
4     String nim;
5     String nama;
6     String kelas;
7     double ipk;
8
9     //Konstruktor default
10    Mahasiswa16() {
11
12    }
13
14    //Konstruktor berparameter (dibuat ada yang nama var parameter inputnya sama ada yang tidak)
15    Mahasiswa16(String nm, String name, String kls, double ip) {
16        nim = nm;
17        nama = name;
18        kelas = kls;
19        ipk = ip;
20    }
21
22    void tampilanInformasi() {
23        System.out.println("Nama : " + nama);
24        System.out.println("NIM : " + nim);
25        System.out.println("Kelas : " + kelas);
26        System.out.println("IPK : " + ipk);
27    }
28
29 }
```
- Bottom Status Bar:** main*, 0 16, BLACKBOX Chat, Add Logs, CyberCoder, Improve Code, Java: Ready, Share Code Link, Generate Commit Message, CRLF, Java, BLACKBOXAI: Open Chat.

3. Buat class MahasiswaBerprestasi seperti di bawah ini!

```

File Edit Selection View Go Run ... ← → 🔍 Praktikum-ASD
EXPLORER OPEN EDITORS 2 unsaved ...
PRAKTIKUM-ASD
> Jobsheet1
> Jobsheet2
> Jobsheet3
> Jobsheet5
J Faktorial.java
J MainFaktorial.java
J MainPangkat.java
J MainSum.java
J Pangkat.java
J Sum.java
Jobsheet6
J Mahasiswa16.java
J MahasiswaBerprestasi16.java U
J Sorting16.java
J SortingMain16.java
> Kuis1
...
@ OUTLINE
@ TIMELINE
@ JAVA PROJECTS
main* ⌂ 0 ⌂ 16 ⌂ BLACKBOX Chat Add Logs CyberCoder Improve Code Java: Ready Share Code Link Generate Commit Message CRLF { } Java 🔍 BLACKBOXAI: Open Chat

```

```

1 package Jobsheet6;
2
3 public class MahasiswaBerprestasi16 {
4     Mahasiswa16 [] listMhs = new Mahasiswa16[5];
5     int idx;
6 }
7

```

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

```

File Edit Selection View Go Run ... ← → 🔍 Praktikum-ASD
EXPLORER OPEN EDITORS 2 unsaved ...
PRAKTIKUM-ASD
> Jobsheet1
> Jobsheet2
> Jobsheet3
> Jobsheet5
J Faktorial.java
J MainFaktorial.java
J MainPangkat.java
J MainSum.java
J Pangkat.java
J Sum.java
Jobsheet6
J Mahasiswa16.java
J MahasiswaBerprestasi16.java U
J Sorting16.java
J SortingMain16.java
> Kuis1
...
@ OUTLINE
@ TIMELINE
@ JAVA PROJECTS
main* ⌂ 0 ⌂ 16 ⌂ BLACKBOX Chat Add Logs CyberCoder Improve Code Java: Ready Share Code Link Generate Commit Message CRLF { } Java 🔍 BLACKBOXAI: Open Chat

```

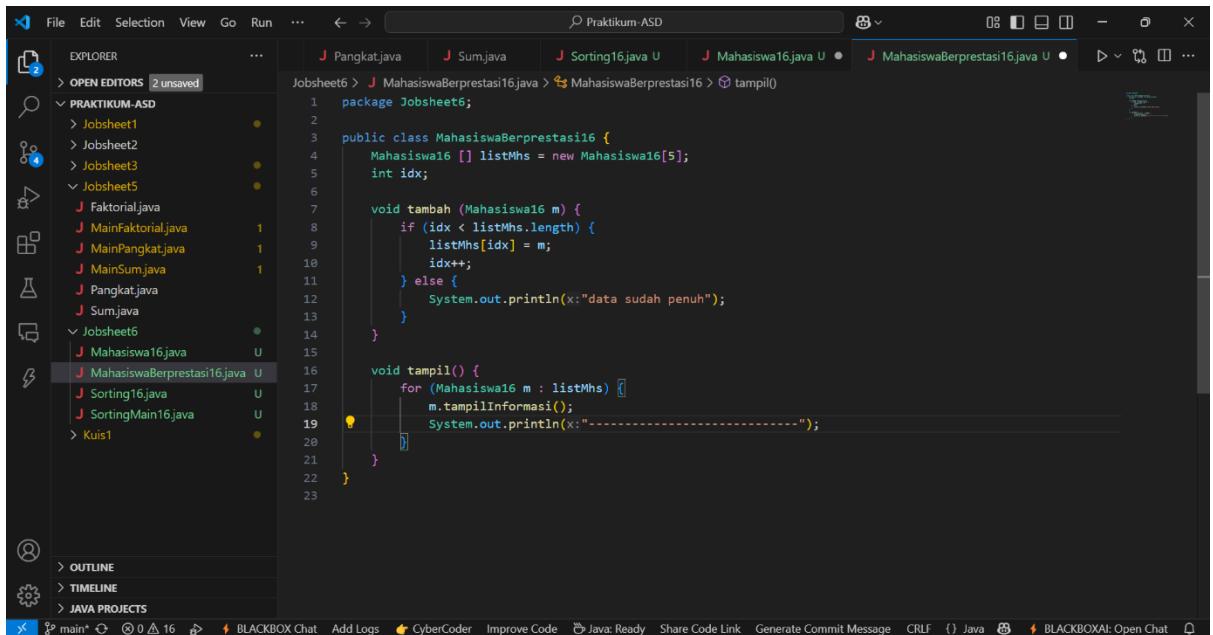
```

1 package Jobsheet6;
2
3 public class MahasiswaBerprestasi16 {
4     Mahasiswa16 [] listMhs = new Mahasiswa16[5];
5     int idx;
6
7     void tambah (Mahasiswa16 m) {
8         if (idx < listMhs.length) {
9             listMhs[idx] = m;
10            idx++;
11        } else {
12            System.out.println("data sudah penuh");
13        }
14    }
15 }
16

```

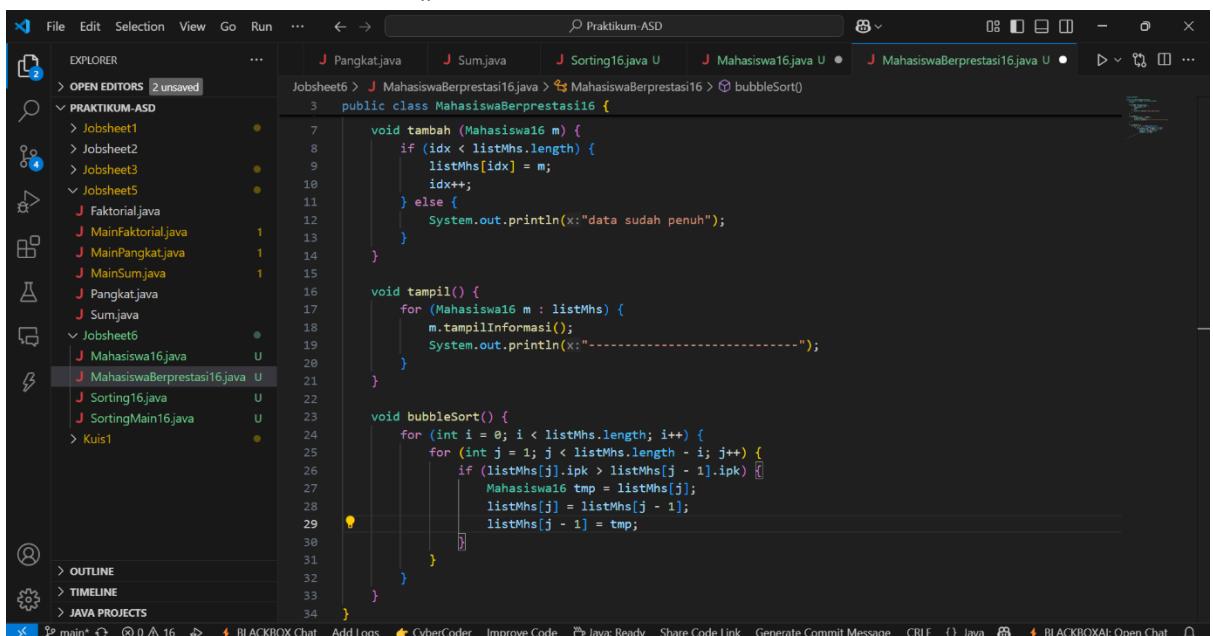
5. Tambahkan method tampil() di dalam class tersebut! Method tampil() digunakan untuk menampilkan semua data mahasiswa-mahasiswa yang ada di dalam class tersebut! Perhatikan penggunaan sintaks for yang agak berbeda dengan for yang

telah dipelajari sebelumnya, meskipun secara konsep sebenarnya mirip.



```
File Edit Selection View Go Run ... ← → 🔍 Praktikum-ASD
EXPLORER OPEN EDITORS 2 unsaved
PRAKTIKUM-ASD
> Jobsheet1
> Jobsheet2
> Jobsheet3
> Jobsheet4
> Jobsheet5
J Faktorial.java
J MainFaktorial.java
J MainPangkat.java
J MainSum.java
J Pangkat.java
J Sum.java
> Jobsheet6
J Mahasiswa16.java
J MahasiswaBerprestasi16.java U
J Sorting16.java
J SortingMain16.java
> Kuis1
...
Jobsheet6 > J MahasiswaBerprestasi16.java > 🔍 MahasiswaBerprestasi16 > 🌐 tampil()
1 package Jobsheet6;
2
3 public class MahasiswaBerprestasi16 {
4     Mahasiswa16 [] listMhs = new Mahasiswa16[5];
5     int idx;
6
7     void tambah (Mahasiswa16 m) {
8         if (idx < listMhs.length) {
9             listMhs[idx] = m;
10            idx++;
11        } else {
12            System.out.println("data sudah penuh");
13        }
14    }
15
16    void tampil() {
17        for (Mahasiswa16 m : listMhs) {
18            m.tampilInformasi();
19        }
20    }
21
22 }
23
```

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



```
File Edit Selection View Go Run ... ← → 🔍 Praktikum-ASD
EXPLORER OPEN EDITORS 2 unsaved
PRAKTIKUM-ASD
> Jobsheet1
> Jobsheet2
> Jobsheet3
> Jobsheet4
> Jobsheet5
J Faktorial.java
J MainFaktorial.java
J MainPangkat.java
J MainSum.java
J Pangkat.java
J Sum.java
> Jobsheet6
J Mahasiswa16.java
J MahasiswaBerprestasi16.java U
J Sorting16.java
J SortingMain16.java
> Kuis1
...
Jobsheet6 > J MahasiswaBerprestasi16.java > 🔍 MahasiswaBerprestasi16 > 🌐 bubbleSort()
3 public class MahasiswaBerprestasi16 {
7     void tambah (Mahasiswa16 m) {
8         if (idx < listMhs.length) {
9             listMhs[idx] = m;
10            idx++;
11        } else {
12            System.out.println("data sudah penuh");
13        }
14    }
15
16    void tampil() {
17        for (Mahasiswa16 m : listMhs) {
18            m.tampilInformasi();
19        }
20        System.out.println("-----");
21    }
22
23    void bubbleSort() {
24        for (int i = 0; i < listMhs.length; i++) {
25            for (int j = 1; j < listMhs.length - i; j++) {
26                if (listMhs[j].ipk > listMhs[j - 1].ipk) {
27                    Mahasiswa16 tmp = listMhs[j];
28                    listMhs[j] = listMhs[j - 1];
29                    listMhs[j - 1] = tmp;
30                }
31            }
32        }
33    }
34 }
```

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.

```

Praktikum-ASD

File Edit Selection View Go Run ... Explorer J Sorting16.java U J Mahasiswa16.java U J MahasiswaBerprestasi16.java U J MahasiswaDemo16.java U ...
J Jobsheet1 J Jobsheet2 J Jobsheet3 J Jobsheet5 J Faktorial.java J MainFaktorial.java J MainPangkat.java J MainSum.java J Pangkat.java J Sum.java J Jobsheet6 J Mahasiswa16.java U J MahasiswaBerprestasi16.java U J MahasiswaDemo16.java U J Sorting16.java U J SortingMain16.java U J Kuis1

1 package Jobsheet6;
2
3 public class MahasiswaDemo16 {
4     public static void main(String[] args) {
5         MahasiswaBerprestasi16 list = new MahasiswaBerprestasi16();
6         Mahasiswa16 m1 = new Mahasiswa16(nm:"123", name:"Zidan", kls:"2A", ip:3.2);
7         Mahasiswa16 m2 = new Mahasiswa16(nm:"124", name:"Ayu", kls:"2A", ip:3.5);
8         Mahasiswa16 m3 = new Mahasiswa16(nm:"125", name:"Sofii", kls:"2A", ip:3.1);
9         Mahasiswa16 m4 = new Mahasiswa16(nm:"126", name:"Sita", kls:"2A", ip:3.9);
10        Mahasiswa16 m5 = new Mahasiswa16(nm:"127", name:"Miki", kls:"2A", ip:3.7);
11
12        list.tambah(m1);
13        list.tambah(m2);
14        list.tambah(m3);
15        list.tambah(m4);
16        list.tambah(m5);
17
18        System.out.println("Data mahasiswa sebelum sorting:");
19        list.tampil();
20
21        System.out.println("\nData Mahasiswa setelah sorting berdasarkan IPK (DESC):");
22        list.bubbleSort();
23        list.tampil();
24    }
25 }

main* 0 16 BLACKBOX Chat Add Logs CyberCoder Improve Code Java: Ready Share Code Link Generate Commit Message CRLF {} Java BLACKBOXAI: Open Chat

```

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

```

Praktikum-ASD

File Edit Selection View Go Run ... PROBLEMS 16 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\zakal\Praktikum-ASD> & 'C:\Program Files\Java\jdk-23\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\zakal\AppData\Roaming\Code\User\workspaceStorage\b499594cf377\bea8a9195ea28df0f41\redhat.ja
va\jdt_ws\Praktikum-d048648\bin' 'Jobsheet6.MahasiswaDemo16'
Data mahasiswa sebelum sorting:
Nama : Zidan
NIM : 123
Kelas : 2A
IPK : 3.2
-----
Nama : Ayu
NIM : 124
Kelas : 2A
IPK : 3.5
-----
Nama : Sofii
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 : Ayu
NIM : 124
Kelas : 2A
IPK : 3.5
-----
Nama : Sofii
NIM : 125
Kelas : 2A
IPK : 3.1
-----
Nama : Miki
NIM : 127
Kelas : 2A
IPK : 3.7
-----
Nama : Zidan
NIM : 123
Kelas : 2A
IPK : 3.2
-----

main* 0 16 BLACKBOX Chat Add Logs CyberCoder Improve Code Java: Ready Share Code Link Generate Commit Message CRLF {} Java BLACKBOXAI: Open Chat

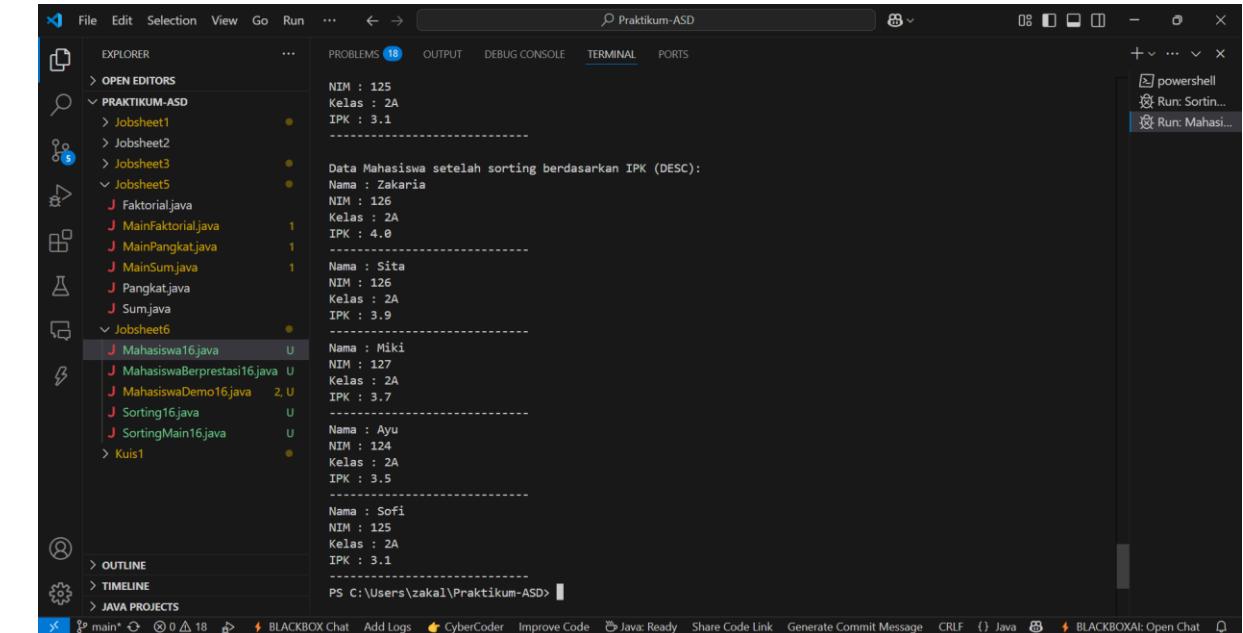
```

Pertanyaan :

1. Perhatikan perulangan di dalam bubbleSort() di bawah ini:
 - a. Mengapa syarat dari perulangan i adalah $i < \text{listMhs.length}-1$?
 - Karena bubble sort bekerja dengan cara Membandingkan dua elemen bertetangga, Menukar posisi kalau urutannya salah
 - b. Mengapa syarat dari perulangan j adalah $j < \text{listMhs.length}-i$?
 - Karena setiap I akan “menyimpan” satu elemen terbesar ke posisi yang benar (dalam sorting descending, elemen dengan IPK tertinggi pindah ke depan)
 - 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

- Jumlah tahap bubble sort = 49

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

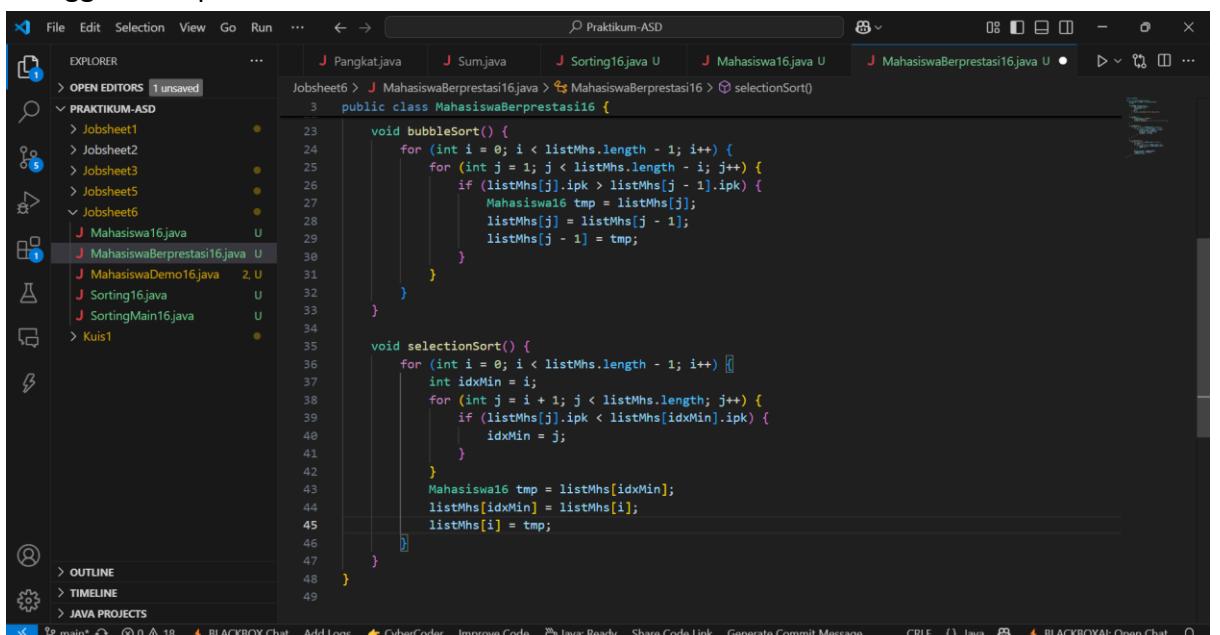


The screenshot shows the VS Code interface with the terminal tab active. The terminal window displays the output of a Java program that sorts student data based on IPK in descending order. The data includes NIM, Name, Kelas, and IPK for several students.

```
NIM : 125
Kelas : 2A
IPK : 3.1
-----
Data Mahasiswa setelah sorting berdasarkan IPK (DESC):
Nama : Zakaria
NIM : 126
Kelas : 2A
IPK : 4.0
-----
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 : Sofi
NIM : 125
Kelas : 2A
IPK : 3.1
```

Mengurutkan Data Mahasiswa Berdasarkan IPK (Selection Sort)

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.



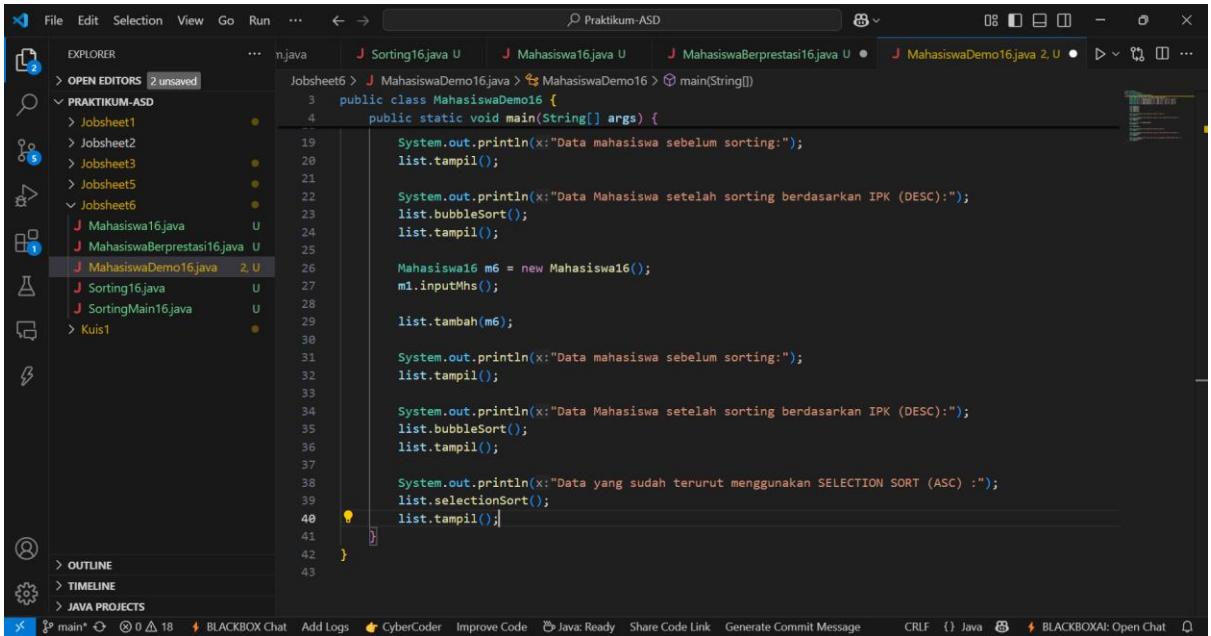
The screenshot shows the VS Code interface with the code editor tab active. The code editor displays the `MahasiswaBerprestasi16.java` file, which contains the implementation of the `selectionSort()` method. The method uses a two-pass approach: first, it finds the minimum value in the array and swaps it with the first element; then, it repeats the process for the remaining elements.

```
public class MahasiswaBerprestasi16 {
    public 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) {
                    Mahasiswa16 tmp = listMhs[j];
                    listMhs[j] = listMhs[j - 1];
                    listMhs[j - 1] = tmp;
                }
            }
        }
    }

    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;
                }
            }
            Mahasiswa16 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,

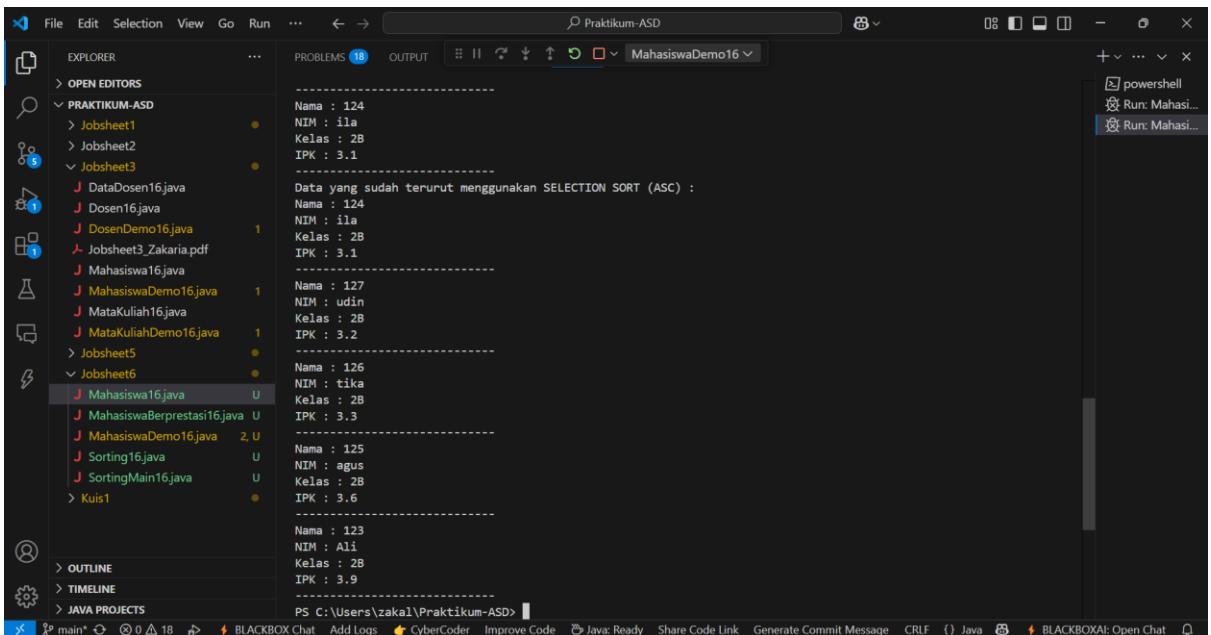
kemudian panggil method tampil() untuk menampilkan data yang sudah diurutkan!



The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows a project named "PRAKTIKUM-ASD" containing several Java files: Jobsheet1, Jobsheet2, Jobsheet3, Jobsheet5, Jobsheet6, Mahasiswa16.java, MahasiswaBerprestasi16.java, MahasiswaDemo16.java, Sorting16.java, SortingMain16.java, and Kuis1.
- Code Editor:** Displays the content of the "MahasiswaDemo16.java" file. The code implements a selection sort algorithm to sort student data based on IPK (Grade Point Average). It includes methods for printing student data before and after sorting, and for inputting student data.
- Bottom Bar:** Shows standard VS Code navigation and status icons.

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



The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows the same project structure as the previous screenshot.
- Output Panel:** Displays the results of running the "MahasiswaDemo16" class. The output shows four sets of student data, each preceded by a dashed line. The data is sorted in ascending order of IPK (Grade Point Average).
- Bottom Bar:** Shows standard VS Code navigation and status icons.

IPK	Nama	NIM	Kelas
3.1	ila	124	2B
3.1	udin	124	2B
3.2	tika	126	2B
3.3	agus	125	2B
3.6	Ali	123	2B

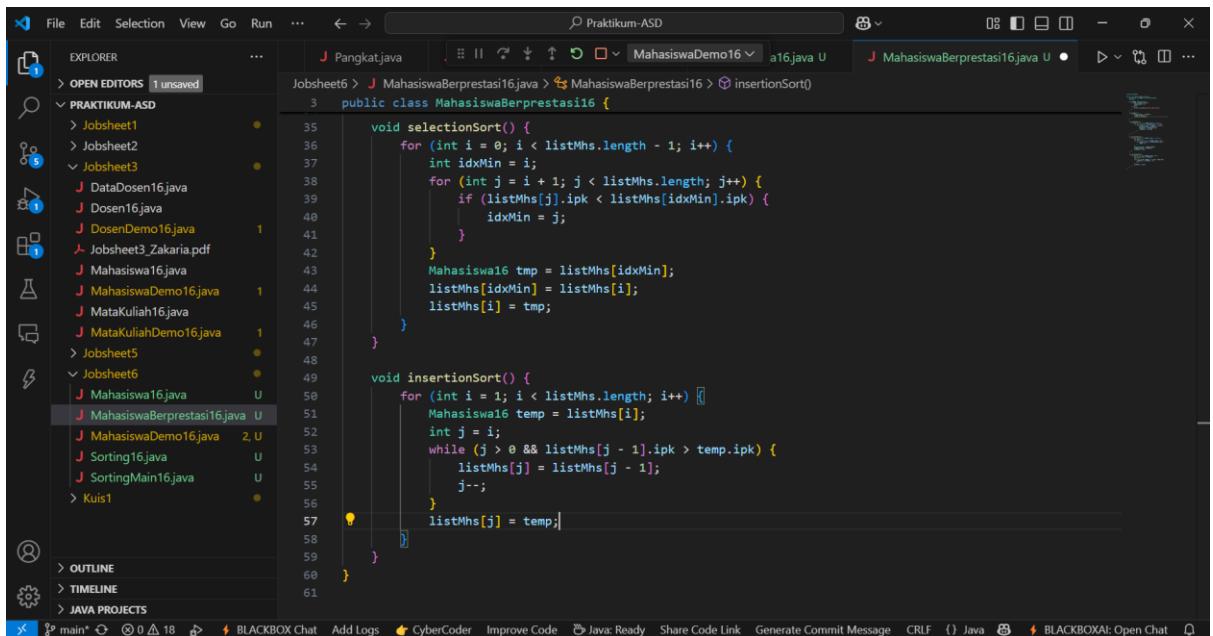
Pertanyaan :

- Di dalam method selection sort, terdapat baris program seperti di bawah ini: Untuk apakah proses tersebut, jelaskan!
 - int idxMin = i;
 - Anggap dulu bahwa elemen ke-i adalah yang paling kecil (minimum) di antara yang belum diurutkan.
 - for (int j = i + 1; j < listMhs.length; j++)
 - Mulai mencari dari elemen setelah posisi ke-i, sampai akhir array.
 - if (listMhs[j].ipk < listMhs[idxMin].ipk)

- Kalau ditemukan elemen dengan IPK lebih kecil, maka update indeks minimum-nya ($\text{idxMin} = j$).
- Setelah keluar dari loop, artinya kita sudah tahu:
Posisi dari elemen dengan IPK terkecil di sisa array.
- Lalu dilakukan proses penukaran (swap) dengan elemen ke- i , agar elemen terkecil itu pindah ke depan.

Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan 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.



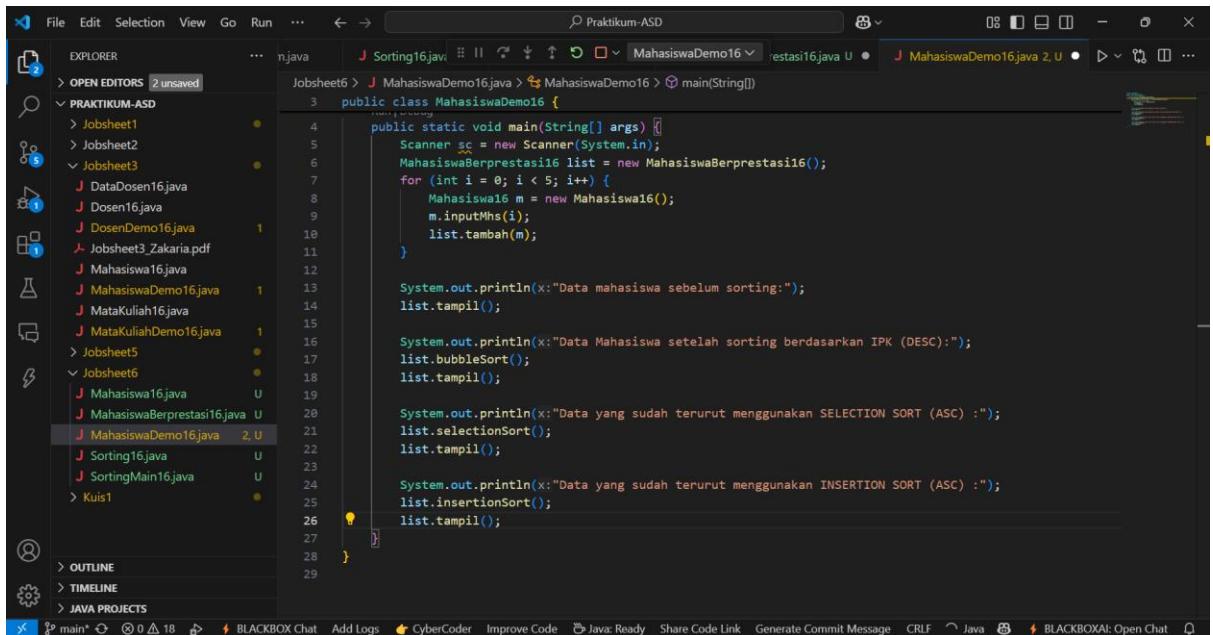
```

File Edit Selection View Go Run ... 🔍 Pangkatjava J MahasiswaBerprestasi16.java a16.java U 🔍 MahasiswaBerprestasi16.java U 🔍 insertionSort()
EXPLORER OPEN EDITORS 1 unsaved PRAKTIKUM-ASD Jobsheet6 > J MahasiswaBerprestasi16.java > MahasiswaBerprestasi16 > insertionSort()
public class MahasiswaBerprestasi16 {
    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;
                }
            }
            Mahasiswa16 tmp = listMhs[idxMin];
            listMhs[idxMin] = listMhs[i];
            listMhs[i] = tmp;
        }
    }
    void insertionSort() {
        for (int i = 1; i < listMhs.length; i++) {
            Mahasiswa16 temp = listMhs[i];
            int j = i;
            while (j > 0 && listMhs[j - 1].ipk > temp.ipk) {
                listMhs[j] = listMhs[j - 1];
                j--;
            }
            listMhs[j] = temp;
        }
    }
}

```

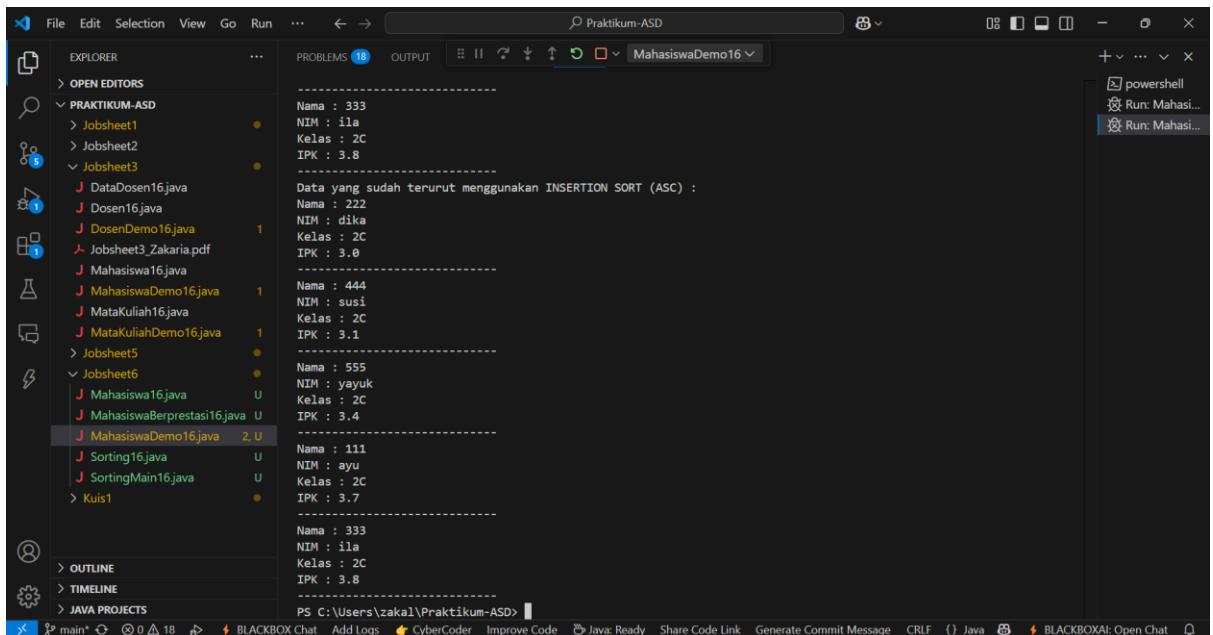
2. Setelah itu, buka kembali class MahasiswaDemo, dan di dalam method main() tambahkan baris program untuk memanggil method insertionSort() dan tampil ()

tersebut!



```
public class MahasiswaDemo16 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        MahasiswaBerprestasi16 list = new MahasiswaBerprestasi16();
        for (int i = 0; i < 5; i++) {
            Mahasiswa16 m = new Mahasiswa16();
            m.inputMhs(i);
            list.tambah(m);
        }
        System.out.println("Data mahasiswa sebelum sorting:");
        list.tampil();
        System.out.println("Data Mahasiswa setelah sorting berdasarkan IPK (DESC):");
        list.bubbleSort();
        list.tampil();
        System.out.println("Data yang sudah terurut menggunakan SELECTION SORT (ASC):");
        list.selectionSort();
        list.tampil();
        System.out.println("Data yang sudah terurut menggunakan INSERTION SORT (ASC):");
        list.insertionSort();
        list.tampil();
    }
}
```

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



```
Nama : 333
NIM : ila
Kelas : 2C
IPK : 3.8
-----
Nama : 444
NIM : susi
Kelas : 2C
IPK : 3.0
-----
Nama : 444
NIM : susi
Kelas : 2C
IPK : 3.0
-----
Nama : 555
NIM : yayuk
Kelas : 2C
IPK : 3.4
-----
Nama : 111
NIM : ayu
Kelas : 2C
IPK : 3.1
-----
Nama : 333
NIM : ila
Kelas : 2C
IPK : 3.8
```

Pertanyaan :

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

Latihan Praktikum



The screenshot shows the Eclipse IDE interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Title Bar:** Praktikum-ASD
- Left Sidebar (Explorer):**
 - OPEN EDITORS (4 unsaved)
 - PRAKTIKUM-ASD
 - Jobsheet1
 - Jobsheet2
 - Jobsheet3
 - DataDosen16.java
 - Dosen16.java
 - DosenDemo16.java
 - Jobsheet3_Zakaria.pdf
 - Mahasiswa16.java
 - MahasiswaDemo16.java
 - MataKuliah16.java
 - MataKuliahDemo16.java
 - Jobsheet5
 - Jobsheet6
 - DataDosen.java
 - Dosen.java
 - Mahasiswa16.java
 - MahasiswaBerprestasi16.java
 - MahasiswaDemo16.java
 - MainDosen.java (selected)
 - Sorting16.java
 - SortingMain16.java
 - Kuis1
 - OUTLINE, TIMELINE, JAVA PROJECTS- Central Area:** The code editor displays Java code for a sorting application. The selected file is MainDosen.java, which contains methods for sorting arrays and printing menu options.
- Bottom Bar:** Improve Code, Java Ready, Share Code Link, Generate Commit Message, CRLF, Java, BLACKBOXAI: Open Chat