

LAPORAN PRAKTIKUM STRUKTUR DATA  
*“Sorting Algorithm 1”*



OLEH :  
EZZA ADDINI  
2311532001

DOSEN PENGAMPU: Dr. WAHYUDI, M. T.

DEPARTEMEN INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS ANDALAS  
PADANG  
2024

## A. PENDAHULUAN

Algoritma pengurutan yang lebih dikenal sebagai algoritma sorting merupakan algoritma untuk meletakkan elemen data kedalam kumpulan data urutan tertentu, sehingga proses pengurutan yang sebelumnya masih tersusun acak menjadi tersusun teratur menurut aturan tertentu. Algoritma ini memungkinkan untuk mengurutkan data secara menaik (ascending) ataupun menurun (descending). Tujuannya adalah untuk membuat pencarian lebih mudah bagi himpunan. Kelebihan data yang terurut adalah bahwa data tersebut mudah dicek apabila ada data yang hilang.

Pada praktikum kali ini, algoritma sorting yang digunakan adalah sebagai berikut:

### a. Insertion Sort

Insertion Sort bekerja dengan cara membangun array yang sudah terurut satu per satu, dengan cara mengambil setiap elemen dan menyisipkannya ke posisi yang benar dalam array yang sudah terurut.

### b. Selection Sort

Selection Sort bekerja dengan cara memilih elemen terkecil dari array dan menukarnya dengan elemen di posisi awal yang belum terurut. Proses ini diulangi untuk setiap posisi dalam array.

### c. Bubble Sort

Bubble Sort adalah algoritma sorting yang paling sederhana. Algoritma ini bekerja dengan cara membandingkan elemen yang berdekatan dan menukar posisi mereka jika berada dalam urutan yang salah. Proses ini diulangi hingga tidak ada lagi elemen yang perlu ditukar.

## B. TUJUAN

Tujuan dari praktikum ini adalah:

1. Membuat program Insertion Sort dengan metode sort.
2. Membuat program Insertion Sort dengan menunjukkan keadaan array sebelum dan sesudah diurutkan.
3. Membuat program Selection Sort.
4. Membuat program Bubble Sort.

### C. LANGKAH KERJA

#### a. Program Insertion Sort dengan Metode Sort

1. Buat package dan class baru, namakan sesuai dengan yang diperintahkan atau yg diinginkan.

```
1 package Pekan5;  
2  
3 public class InsertionSort {
```

2. Buat sintaks untuk mengurutkan array integer.

```
void sort(int arr[]) {  
    int n = arr.length;  
    for (int i = 1; i < n; ++i) {  
        int key = arr[i];  
        int j = i - 1;  
  
        while (j >= 0 && arr[j] > key) {  
            j = j - 1;  
        }  
        arr[j + 1] = key;
```

3. Buat metode untuk mencetak elemen-elemen dari sebuah array integer ke konsol output.

```
static void printArray(int arr[]) {  
    int n = arr.length;  
    for (int i = 0; i < n; ++i)  
        System.out.print(arr[i] + " ");  
  
    System.out.println();
```

4. Insialisasi array.

```
public static void main(String args[]) {  
    int arr[] = {3, 10, 4, 6, 8, 9, 7, 2, 1, 5};
```

5. Buat sintaks memanggil metode sort pada instance ob, yang mengurutkan array arr di tempat menggunakan Insertion Sort.

```
InsertionSort ob = new InsertionSort();  
ob.sort(arr);
```

6. Buat sintaks untuk mencetak elemen-elemen dari array arr yang sudah diurutkan ke konsol output.

```
printArray(arr);
```

7. Jalankan program.

```
Console ×  
<terminated> InsertionSort [Java  
1 4 4 6 7 9 7 2 1 5
```

#### b. Membuat Program Insertion Sort 2

1. Buat class baru dan namakan sesuai perintah atau yang diinginkan.

```
package Pekan5;  
  
public class InsertionSort2 {
```

2. Buat metode untuk mendeklarasikan tipe data array.

```
public static void insertionSort(int[] arr) {
    int n = arr.length;
```

3. Buat sintaks perulangan for untuk memastikan setiap elemen dari array ditempatkan pada posisi yang benar dalam sub-array yang sudah diurutkan.

```
for (int i = 1; i < n; i++) {
    int key = arr[i];
    int j = i - 1;
```

4. Buat sintaks perulangan while untuk memastikan bahwa setiap elemen key ditempatkan pada posisi yang benar dalam sub-array yang sudah diurutkan, dengan menggeser elemen-elemen yang lebih besar satu posisi ke kanan.

```
while (j >= 0 && arr[j] > key) {
    arr[j + 1] = arr[j];
    j--;
}
arr[j + 1] = key;
```

5. Inisialisasi array.

```
public static void main(String[] args) {
    int arr[] = {3, 10, 4, 6, 8, 9, 7, 2, 1, 5};
    int n = arr.length;
```

6. Buat sintaks untuk mencetak array sebelum pengurutan, memanggil metode insertionSort untuk mengurutkan array, dan kemudian mencetak array setelah pengurutan.

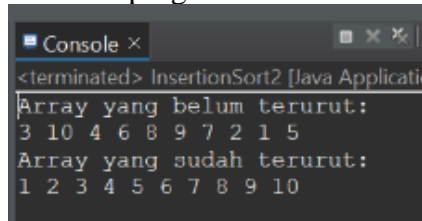
```
System.out.printf("Array yang belum terurut: \n");
for (int i = 0; i < n; i++) {
    System.out.print(arr[i] + " ");
}

System.out.println("");
insertionSort(arr);

System.out.printf("Array yang sudah terurut: \n");
for (int i = 0; i < n; i++) {
    System.out.print(arr[i] + " ");
}

System.out.println();
```

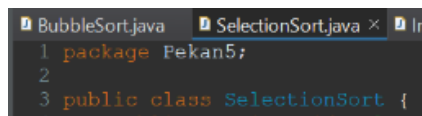
7. Jalankan program.



```
Console x
<terminated> InsertionSort2 [Java Applicati
Array yang belum terurut:
3 10 4 6 8 9 7 2 1 5
Array yang sudah terurut:
1 2 3 4 5 6 7 8 9 10
```

### c. Program Selection Sort

1. Buat class baru dan namakan sesuai perintah atau yang diinginkan.



```
BubbleSort.java SelectionSort.java x
1 package Pekan5;
2
3 public class SelectionSort {
4     public static void main()
```

2. Deklarasikan metode dan panjang array yang akan digunakan.

```
public static void selectionSort (int[] arr) {  
    int n = arr.length;
```

3. Buat sintaks untuk mencari indeks minimum dalam subarray yang belum diurutkan.

```
    for (int i = 0; i < n; i++) {  
        int minIndex = i;  
        for (int j = i + 1; j < n; j++) {  
            if (arr[j] < arr[minIndex]) {  
                minIndex = j;  
            }  
        }  
    }
```

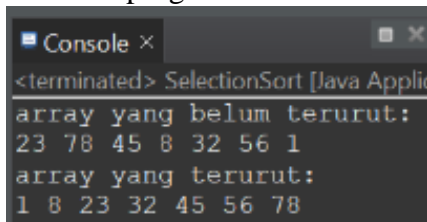
4. Buat sintaks untuk menemukan nilai minimum dan ditukar dengan elemen pertama dari subarray tersebut.

```
    int temp = arr[i];  
    arr[i] = arr[minIndex];  
    arr[minIndex] = temp;  
}
```

5. Buat main method untuk mengurutkan array.

```
public static void main (String[] args) {  
    int arr[] = {23, 78, 45, 8, 32, 56, 1};  
    int n = arr.length;  
    System.out.printf("array yang belum terurut: \n");  
    for (int i = 0; i < n; i++)  
        System.out.print(arr[i] + " ");  
    System.out.println("");  
    selectionSort(arr);  
    System.out.printf("array yang terurut: \n");  
    for (int i = 0; i < n; i++)  
        System.out.print(arr[i] + " ");  
    System.out.println("");  
}
```

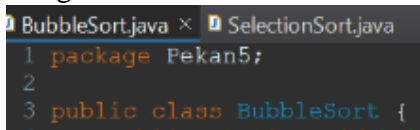
6. Jalankan program.



```
Console x  
<terminated> SelectionSort [Java Appli  
array yang belum terurut:  
23 78 45 8 32 56 1  
array yang terurut:  
1 8 23 32 45 56 78
```

d. Program Bubble Sort

1. Buatlah class baru dan namakan sesuai perintah atau yang diinginkan.



```
BubbleSort.java x SelectionSort.java  
1 package Pekan5;  
2  
3 public class BubbleSort {  
4     // ...  
5 }
```

2. Deklarasikan metode dan panjang array yang akan digunakan.

```
public static void bubbleSort (int[] arr) {  
    int n = arr.length;
```

3. Buat perulangan for untuk memindahkan elemen terbesar ke posisi yang benar secara bertahap, hingga seluruh array terurut.

```
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n - i - 1; j++) {
        if (arr[j] > arr[j + 1]) {
            int temp = arr[j];
            arr[j] = arr[j + 1];
            arr[j + 1] = temp;
        }
    }
}
```

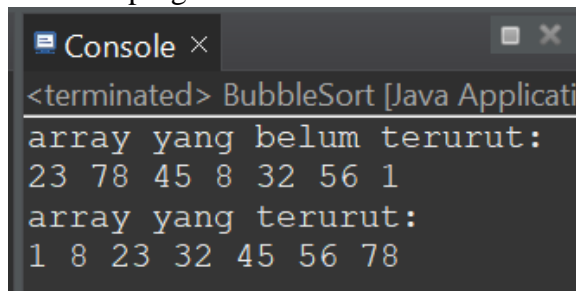
4. Inisialisasi isi array dan tentukan panjangnya.

```
public static void main (String[] args) {
    int arr[] = {23, 78, 45, 8, 32, 56, 1};
    int n = arr.length;
}
```

5. Buat sintaks untuk mencetak array sebelum dan sesudah diurutkan, kemudian buat algoritma bubble sort-nya.

```
public static void main (String[] args) {
    int arr[] = {23, 78, 45, 8, 32, 56, 1};
    int n = arr.length;
    System.out.printf("array yang belum terurut: \n");
    for (int i = 0; i < n; i++)
        System.out.print(arr[i] + " ");
    System.out.println(""); //minMaxSelectionSort (arr, n);
    bubbleSort(arr);
    System.out.printf("array yang terurut: \n");
    for (int i = 0; i < n; i++)
        System.out.print(arr[i] + " ");
    System.out.println("");
}
```

6. Jalankan program.



```
<terminated> BubbleSort [Java Applicati
array yang belum terurut:
23 78 45 8 32 56 1
array yang terurut:
1 8 23 32 45 56 78
```

#### D. KESIMPULAN

Pada praktikum ini, kita telah mempelajari dan mengimplementasikan tiga jenis algoritma sorting: Insertion Sort, Selection Sort, dan Bubble Sort. Masing-masing algoritma memiliki pendekatan yang berbeda dalam mengurutkan data:

1. Insertion Sort efektif untuk dataset kecil dan hampir terurut, karena secara bertahap membangun array yang sudah terurut dengan menyisipkan elemen pada posisi yang benar. Algoritma ini memiliki kompleksitas waktu rata-rata dan terburuk  $O(n^2)$ .
2. Selection Sort memilih elemen terkecil dari bagian array yang belum terurut dan menukarnya dengan elemen di posisi awal yang belum terurut. Meskipun sederhana dan mudah dipahami, Selection Sort juga memiliki kompleksitas waktu  $O(n^2)$  dan tidak efisien untuk dataset besar.
3. Bubble Sort bekerja dengan cara membandingkan dan menukar elemen yang berdekatan hingga tidak ada lagi elemen yang perlu ditukar. Algoritma ini adalah yang paling sederhana di antara ketiganya, namun memiliki performa yang buruk dengan kompleksitas waktu  $O(n^2)$  dan hanya cocok untuk dataset yang sangat kecil atau hampir terurut.

Melalui praktikum ini, kita dapat memahami dasar-dasar dari berbagai algoritma sorting dan mempertimbangkan efisiensi serta aplikasi praktis mereka. Ketiga algoritma ini memberikan landasan yang baik untuk memahami konsep pengurutan sebelum mempelajari algoritma yang lebih kompleks dan efisien.