

Worksheet pertemuan 12 – 1
Algoritma dan Struktur Data
Algoritma Pengurutan – Bubble Sort

NIM: 20523164

Nama: Fajrun Shubhi

A. Membuat Folder Untuk Menyimpan Hasil Praktikum

1. Siapkan folder kosong dengan nama menggunakan NIM masing-masing. Jika folder NIM pada pertemuan sebelumnya mau dimanfaatkan, jangan lupa pindahkan dulu isinya ke folder lain sebagai arsip.
2. Folder ini akan dijadikan tempat untuk menyimpan semua pdf dari worksheet ini beserta fail praktikum lainnya.

B. Membuat dan menjalankan class BubbleSort

1. Buatlah sebuah *class* dengan nama BubbleSort
2. Kemudian salin dan tempel kode program di bawah ini

```
public class BubleSort {

    public void sorting(int array[]) {
        int n = 1;

        while(n < array.length){
            for (int i = 0; i < (array.length - 1); i++){
                if (array[i] > array[i+1]){
                    int temp = array[i];
                    array[i] = array[i+1];
                    array[i+1] = temp;
                }
            }
            n++;
        }
    }

    public static void main(String[] args) {
        int data[] = {35,13,26,10,39,40};
        BubleSort b = new BubleSort();

        System.out.println("Array awal: ");
        for (int i=0; i<data.length; i++){
            System.out.print(data[i]+" ");
        }
        System.out.println("");
    }
}
```

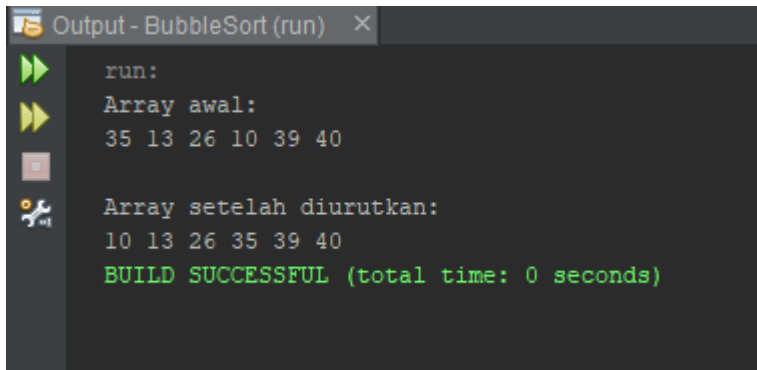
```

        b.sorting(data);

        System.out.println("");
        System.out.println("Array setelah diurutkan: ");
        for (int i=0; i<data.length; i++){
            System.out.print(data[i]+" ");
        }
        System.out.println("");
    }
}

```

3. Jalankan *class* BubbleSort di atas lalu letakkan hasil tangkapan layarnya di kolom di bawah ini



```

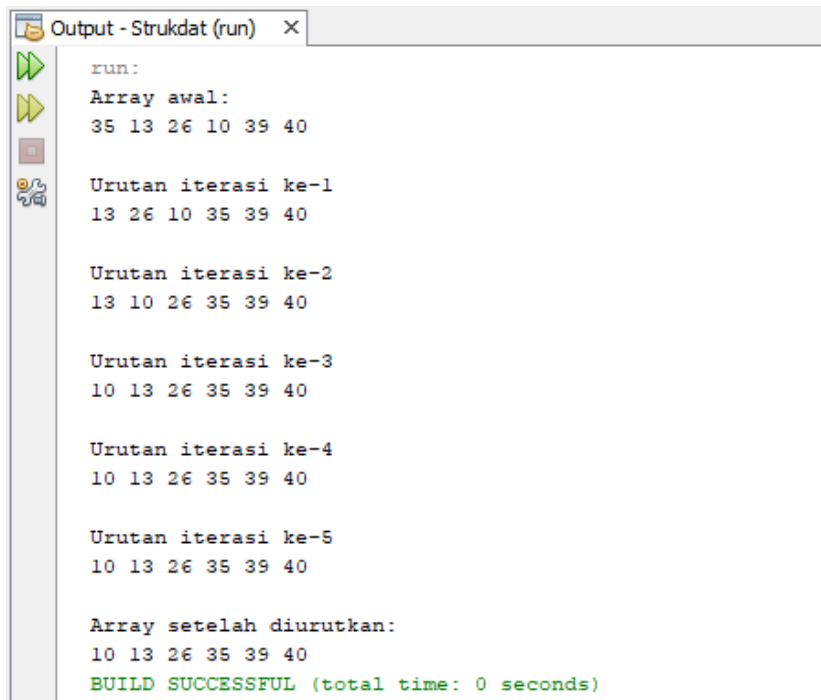
run:
Array awal:
35 13 26 10 39 40

Array setelah diurutkan:
10 13 26 35 39 40
BUILD SUCCESSFUL (total time: 0 seconds)

```

B. Modifikasi class BubbleSort Bagian 1

1. Lakukan pembaruan *class* BubbleSort agar dapat menampilkan urutan array di setiap iterasi. Contoh tampilannya seperti di bawah ini.



```
run:
Array awal:
35 13 26 10 39 40

Urutan iterasi ke-1
13 26 10 35 39 40

Urutan iterasi ke-2
13 10 26 35 39 40

Urutan iterasi ke-3
10 13 26 35 39 40

Urutan iterasi ke-4
10 13 26 35 39 40

Urutan iterasi ke-5
10 13 26 35 39 40

Array setelah diurutkan:
10 13 26 35 39 40
BUILD SUCCESSFUL (total time: 0 seconds)
```

2. Berdasarkan contoh gambar di atas pada iterasi ke berapakah semua data sudah terurut?

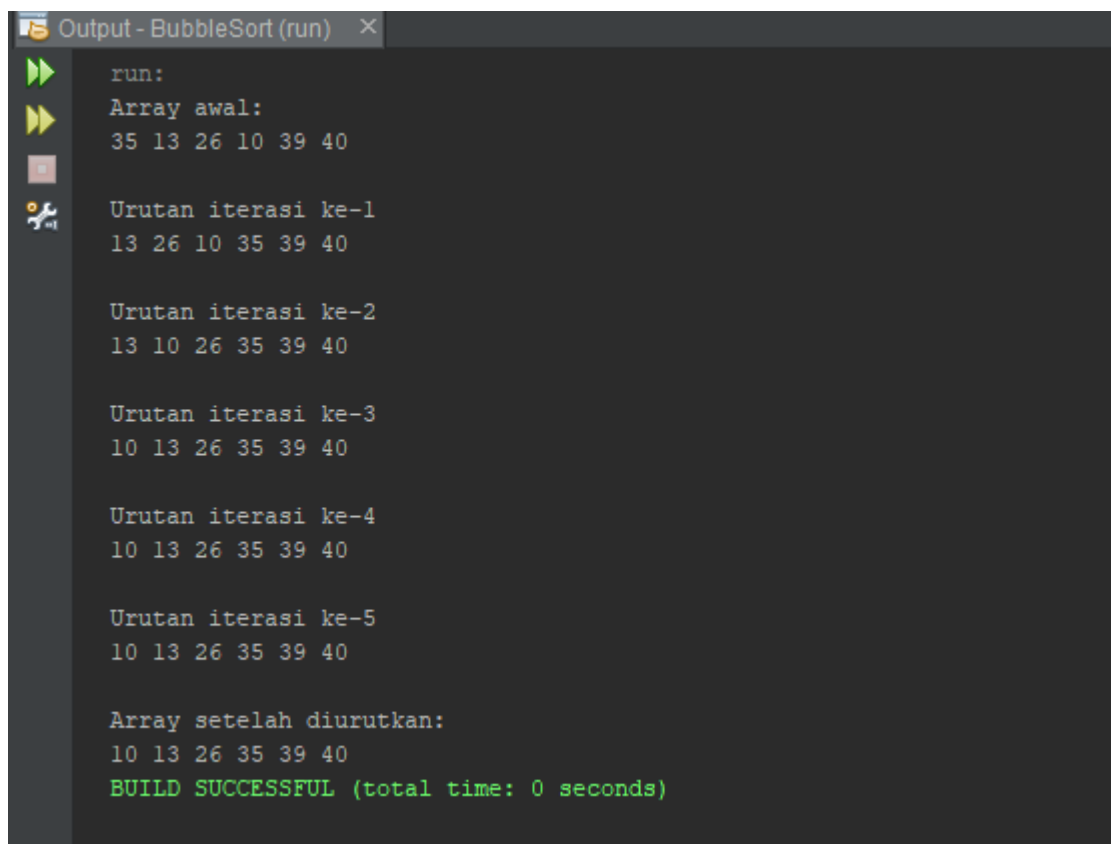
Pada iterasi ke-3 semua data sudah terurut.

3. Salin tempel isi *class* BubbleSort yang sudah Anda perbarui ke kolom di bawah ini

```
// berikut method sorting yang sudah diperbarui.

public void sorting(int array[]) {
    int n = 1;
    while(n < array.length){
        for (int i = 0; i < (array.length - 1); i++){
            if (array[i] > array[i+1]){
                int temp = array[i];
                array[i] = array[i+1];
                array[i+1] = temp;
            }
        }
        System.out.println("\nUrutan iterasi ke-" + n);
        for (int i=0; i<array.length; i++){
            System.out.print(array[i]+" ");
        }
        System.out.println("");
        n++;
    }
}
```

4. Jalankan *class* BubbleSort yang sudah Anda perbarui lalu letakkan hasil tangkapan layarnya di kolom di bawah ini



```
run:
Array awal:
35 13 26 10 39 40

Urutan iterasi ke-1
13 26 10 35 39 40

Urutan iterasi ke-2
13 10 26 35 39 40

Urutan iterasi ke-3
10 13 26 35 39 40

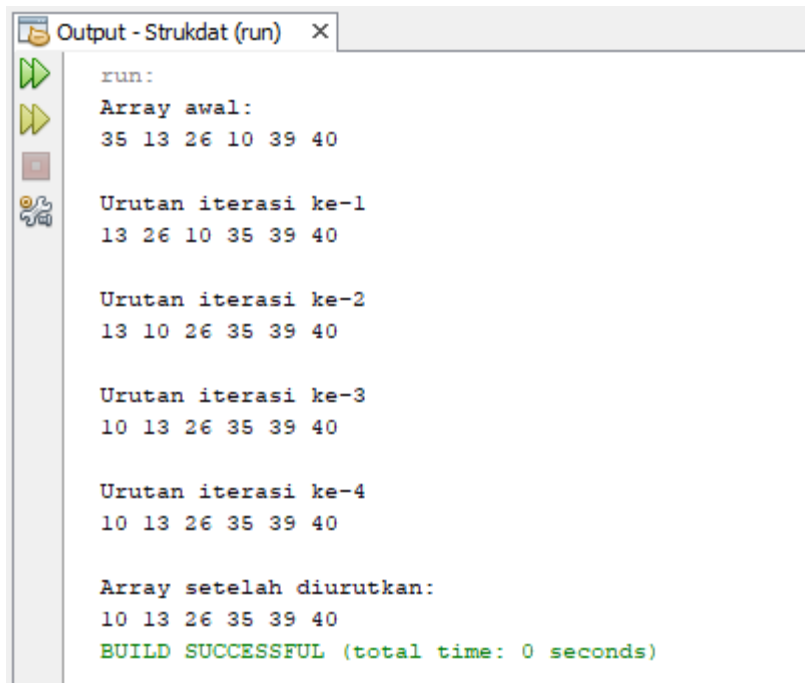
Urutan iterasi ke-4
10 13 26 35 39 40

Urutan iterasi ke-5
10 13 26 35 39 40

Array setelah diurutkan:
10 13 26 35 39 40
BUILD SUCCESSFUL (total time: 0 seconds)
```

C. Modifikasi class BubbleSort Bagian 2

1. Silakan baca lagi *slide* terakhir materi Pertemuan 12.1 - Algoritma Pengurutan - Bubble Sort tentang kendala yang terdapat pada algoritme Bubble Sort
2. Lakukan pembaruan *class* BubbleSort agar dapat mengatasi kendala tersebut.
 - Tambahkan *suatu variabel* yang akan membantu mengecek apakah ada pertukaran yang terjadi atau tidak.
 - Jika tidak ada pertukaran yang terjadi dalam satu iterasi, proses perulangan akan berhenti.
 - Dengan artian bahwa semua data sudah terurut pada iterasi sebelum iterasi terakhir yang tidak terjadi pertukaran di dalamnya
3. Hasil akhir ketika *class* BubbleSort dijalankan akan seperti gambar di bawah ini



```
run:
Array awal:
35 13 26 10 39 40

Urutan iterasi ke-1
13 26 10 35 39 40

Urutan iterasi ke-2
13 10 26 35 39 40

Urutan iterasi ke-3
10 13 26 35 39 40

Urutan iterasi ke-4
10 13 26 35 39 40

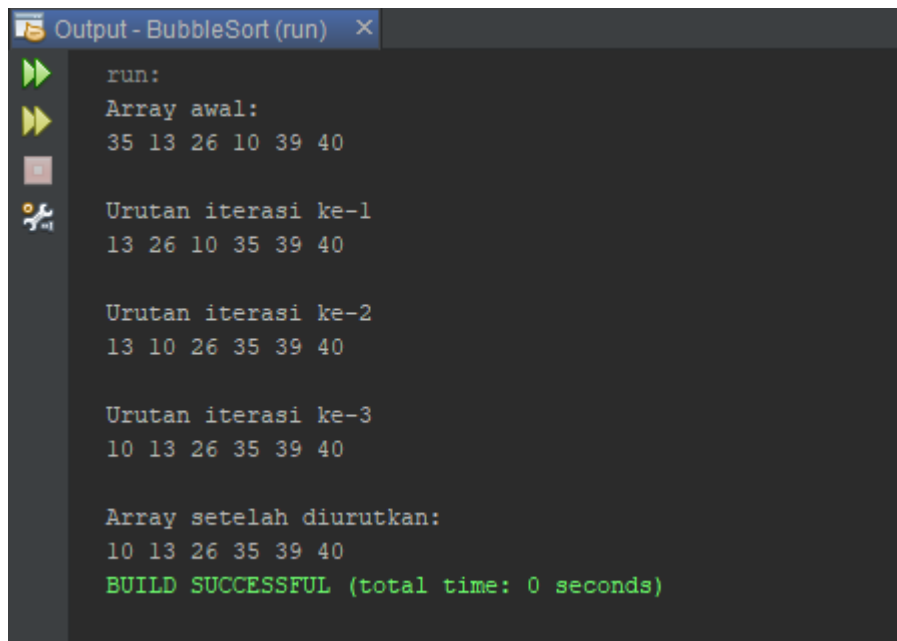
Array setelah diurutkan:
10 13 26 35 39 40
BUILD SUCCESSFUL (total time: 0 seconds)
```

4. Dari gambar di atas dapat kita lihat bahwa semua data sudah terurut setelah iterasi ke-3 berakhir sehingga pada iterasi ke-4 tidak terjadi pertukaran data lagi
5. Proses iterasi berhenti pada iterasi ke-4 karena sudah tidak terjadi pertukaran lagi.
6. Salin tempel isi *class* BubbleSort yang sudah Anda perbarui ke kolom di bawah ini

```
public void sorting(int array[]) {
    int n = 1;
    while(n < array.length){
        boolean tukar = false;
        for (int i = 0; i < (array.length - 1); i++){
            if (array[i] > array[i+1]){
                int temp = array[i];
                array[i] = array[i+1];
                array[i+1] = temp;
                tukar = true;
            }
        }
        if (!tukar){
            break;
        }

        System.out.println("\nUrutan iterasi ke-" + n);
        for (int i=0; i<array.length; i++){
            System.out.print(array[i]+" ");
        }
        System.out.println("");
        n++;
    }
}
```

6. Jalankan *class* BubbleSort yang sudah Anda perbarui lalu letakkan hasil tangkapan layarnya di kolom di bawah ini



```
run:
Array awal:
35 13 26 10 39 40

Urutan iterasi ke-1
13 26 10 35 39 40

Urutan iterasi ke-2
13 10 26 35 39 40

Urutan iterasi ke-3
10 13 26 35 39 40

Array setelah diurutkan:
10 13 26 35 39 40
BUILD SUCCESSFUL (total time: 0 seconds)
```

***Catatan**

- Jangan lupa simpan juga fail worksheet ini (yang sudah diisi) sebagai fail pdf di folder NIM anda.
- Sertakan juga fail **BubbleSort.java** di dalam folder yang Anda gunakan
- Kompres folder ini sebagai fail ZIP kemudian kumpulkan di classroom atau ruang pengumpulan lain di kelas masing-masing.