

# **LAPORAN PRAKTIKUM STRUKTUR DATA**

## **Jobsheet 8 Bubble Sort and Insertion Sort**



**Disusun Oleh:**

**Fitri Walidi: 22343021**

**Dosen Pengampu:**

**Randi Proska Sandra S.Pd., M.Sc.**

**PROGRAM STUDI S1 INFORMATIKA (NK)**

**JURUSAN TEKNIK ELEKTRONIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS NEGERI PADANG**

**2023**

## Tugas

Carilah contoh aplikasi yang mengimplementasikan *bubble sort* dan *insertion sort* serta jelaskan bagaimana aplikasi tersebut bekerja sesuai dengan prinsip kedua metode sorting tersebut!

### 1. Bubble Sort

kode:

```
//Dibuat_oleh_Fitri-Waldi_NIM-22343021
```

```
#include <iostream>
```

```
using namespace std;
```

```
void bubbleSort(int arr[], int n) {  
    for (int i = 0; i < n - 1; i++) {  
        for (int j = 0; j < n - i - 1; j++) {  
            if (arr[j] > arr[j + 1]) {  
                // Swap arr[j] and arr[j+1]  
                int temp = arr[j];  
                arr[j] = arr[j + 1];  
                arr[j + 1] = temp;  
            }  
        }  
    }  
}
```

```
void printArray(int arr[], int n) {  
    for (int i = 0; i < n; i++) {  
        cout << arr[i] << " ";  
    }  
    cout << endl;  
}
```

```
int main() {  
    int arr[] = { 64, 34, 25, 12, 22, 11, 90};  
    int n = sizeof(arr) / sizeof(arr[0]);  
  
    cout << "Array sebelum sorting: ";  
    printArray(arr, n);  
  
    bubbleSort(arr, n);
```

```
cout << "Array setelah Bubble Sort: ";  
printArray(arr, n);  
  
return 0;  
}
```

Output:

```
Output Clear  
/tmp/C5cZnmfGQb.o  
Array sebelum sorting: 64 34 25 12 22 11 90  
Array setelah Bubble Sort: 11 12 22 25 34 64 90
```

Penjelasan:

Program di atas adalah implementasi Bubble Sort dalam bahasa C++. Bubble Sort adalah algoritma pengurutan sederhana yang bekerja dengan membandingkan dan menukar pasangan elemen secara berurutan hingga seluruh array terurut.

Pada program tersebut, terdapat dua fungsi utama:

1. **bubbleSort**: Fungsi ini menerima array dan jumlah elemen sebagai argumen. Fungsi ini menggunakan perulangan nested (perulangan di dalam perulangan) untuk membandingkan dan menukar pasangan elemen bersebelahan jika urutannya tidak benar. Proses ini diulang sebanyak **n - 1** kali, di mana **n** adalah jumlah elemen dalam array. Setiap iterasi, elemen terbesar akan naik ke posisi yang tepat.
2. **printArray**: Fungsi ini menerima array dan jumlah elemen sebagai argumen. Fungsi ini digunakan untuk mencetak elemen-elemen array dalam satu baris.

Di dalam fungsi **main**, terdapat langkah-langkah berikut:

1. Sebuah array awal dideklarasikan dengan nilai yang belum terurut.
2. Jumlah elemen dalam array dihitung menggunakan **sizeof** dan kemudian dibagi dengan **sizeof** elemen pertama dalam array.
3. Array awal dicetak.
4. Fungsi **bubbleSort** dipanggil untuk mengurutkan array.
5. Array yang sudah terurut dicetak.

Dengan menjalankan program tersebut, kita akan melihat array awal dan array yang sudah terurut dengan menggunakan Bubble Sort. Bubble Sort bekerja dengan membandingkan dan menukar elemen-elemen bersebelahan secara berulang hingga seluruh array terurut dengan benar.

## 2. Insertion Sort

Kode:

```
////Dibuat_oleh_Fitri-Waldi_NIM-22343021

#include <iostream>
using namespace std;

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

        // Memindahkan elemen-elemen yang lebih besar dari key ke posisi yang tepat
        while (j >= 0 && arr[j] > key) {
            arr[j + 1] = arr[j];
            j--;
        }

        // Menempatkan key ke posisi yang tepat
        arr[j + 1] = key;
    }
}

void printArray(int arr[], int n) {
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
}

int main() {
    int arr[] = { 64, 34, 25, 12, 22, 11, 90};
    int n = sizeof(arr) / sizeof(arr[0]);

    cout << "Array sebelum sorting: ";
```

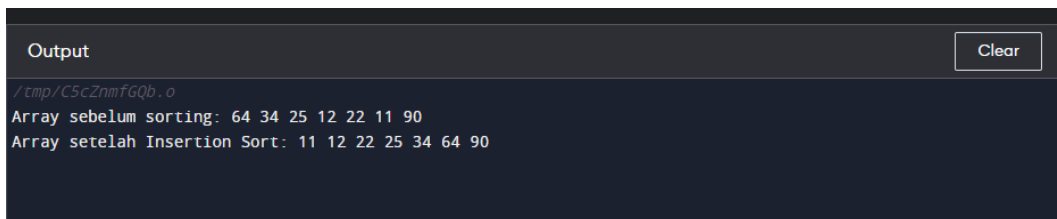
```
printArray(arr, n);

insertionSort(arr, n);

cout << "Array setelah Insertion Sort: ";
printArray(arr, n);

return 0;
}
```

Output:



```
Output
/tmp/C5cZnmfGQb.o
Array sebelum sorting: 64 34 25 12 22 11 90
Array setelah Insertion Sort: 11 12 22 25 34 64 90
```

Penjelasan:

Program di atas mengimplementasikan algoritma Insertion Sort untuk mengurutkan array secara ascending (dari terkecil ke terbesar). Berikut penjelasan singkatnya:

1. Fungsi **insertionSort**: Mengiterasi melalui elemen-elemen array, dimulai dari indeks 1. Pada setiap iterasi, elemen saat ini disimpan dalam variabel **key**, dan dilakukan perulangan mundur untuk memindahkan elemen-elemen yang lebih besar dari **key** ke posisi yang tepat. Akhirnya, **key** ditempatkan pada posisi yang sesuai.
2. Fungsi **printArray**: Mencetak elemen-elemen array dalam satu baris.
3. Di dalam fungsi **main**:
  - Sebuah array awal dideklarasikan dengan nilai yang belum terurut.
  - Jumlah elemen dalam array dihitung.
  - Array awal dicetak.
  - Fungsi **insertionSort** dipanggil untuk mengurutkan array.
  - Array yang sudah terurut dicetak.

Dengan menggunakan Insertion Sort, program akan mengurutkan array secara bertahap dengan membandingkan dan memindahkan elemen-elemen hingga seluruh array terurut secara benar.