

## Tugas Job Sheet 8 Bubble and Insertion Sort\_Praktikum Struktur data

### Muhamad Akbar Fauzan\_23343075

Berikut adalah contoh Aplikasi mengurutkan array yang acak menjadi berurutan dengan menggunakan Bubble and Insertion Sort

```
// js 08 T 01.c
// created by 23343075_muhamad akbar fauzan
// program mengurutkan array dengna Bubble and Insertion Sort

#include <stdio.h>

// Fungsi untuk menampilkan array
void printArray(int arr[], int size) {
    for (int i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

// Bubble Sort
void bubbleSort(int arr[], int size) {
    for (int i = 0; i < size - 1; i++) {
        for (int j = 0; j < size - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                // Tukar elemen
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}

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

        // Pindahkan elemen yang lebih besar dari key ke posisi satu di depannya
        while (j >= 0 && arr[j] > key) {
            arr[j + 1] = arr[j];
            j = j - 1;
        }
    }
}
```

```

        arr[j + 1] = key;
    }
}

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

    printf("Array sebelum diurutkan:\n");
    printArray(arr, size);

    printf("\nMenggunakan Bubble Sort:\n");
    bubbleSort(arr, size);
    printArray(arr, size);

    printf("\nMenggunakan Insertion Sort:\n");
    insertionSort(arr, size);
    printArray(arr, size);

    return 0;
}

```

```

Array sebelum diurutkan:
64 34 25 12 22 11 90

Menggunakan Bubble Sort:
11 12 22 25 34 64 90

Menggunakan Insertion Sort:
11 12 22 25 34 64 90

-----
Process exited after 0.04943 seconds with return value 0
Press any key to continue . . . |

```

## Penjelasan

### Bubble Sort:

Bubble sort bekerja dengan membandingkan setiap pasang elemen adjacent dalam array dan menukarnya jika mereka tidak dalam urutan yang benar. Proses ini diulangi untuk setiap pasangan elemen hingga seluruh array terurut. Pada setiap iterasi, elemen terbesar akan 'naik ke atas' seperti gelembung dalam air, hingga semua elemen terurut.

### Insertion Sort:

Insertion sort bekerja dengan membagi array menjadi dua bagian: subarray terurut dan subarray tidak terurut. Pada setiap iterasi, satu elemen dari subarray tidak terurut diambil dan dimasukkan ke dalam posisi yang sesuai dalam subarray terurut. Proses ini berlanjut hingga semua elemen disisipkan dan array menjadi terurut.

Kedua metode sorting ini memiliki kompleksitas waktu  $O(n^2)$ , namun insertion sort cenderung lebih efisien daripada bubble sort dalam kasus terbaiknya karena memiliki jumlah iterasi yang lebih sedikit jika array sudah dalam keadaan hampir terurut.