

TUGAS JOBSHEET 8



Disusun Oleh:

Qadrian Zakhri

23343081

Dosen pengampu :

Randi Proska Sandra, M.Sc

PROGRAM STUDI
INFORMATIKA(NK) DEPARTEMEN
ELEKTRONIKA FAKULTAS TEKNIK
UNIVERSITAS NEGERI PADANG

Source Code

```
#include <stdio.h>

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]) {
                int
temp = arr[j];
                arr[j]
= arr[j+1];

arr[j+1] = temp;
            }
        }
    }
}

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

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

```
        arr[j + 1] =
key;
    }
}
```

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

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

```
    printf("Original
array for Bubble Sort:
\n");
```

```
printArray(bubbleSortA
rray, n);
```

```
bubbleSort(bubbleSortA
rray, n);
```

```

        printf("Sorted
array using Bubble
Sort: \n");

printArray(bubbleSortA
rray, n);

    int
insertionSortArray[] =
{64, 34, 25, 12, 22,
11, 90};

    n =
sizeof(insertionSortAr
ray)/sizeof(insertionS
ortArray[0]);

    printf("Original
array for Insertion
Sort: \n");

printArray(insertionSo
rtArray, n);

insertionSort(insertio
nSortArray, n);

    printf("Sorted
array using Insertion
Sort: \n");

printArray(insertionSo
rtArray, n);

    return 0;
}

```

Implementasi Bubble

Sort dan Insertion Sort

1. Bubble Sort

- Loop luar berjalan `n-1` kali, di mana `n` adalah panjang array.
- Loop dalam berjalan dari awal array hingga elemen terakhir dari sub-array yang belum diurutkan.
- Elemen-elemen yang tidak dalam urutan yang benar ditukar.

2. Insertion Sort

- Loop utama berjalan dari elemen kedua hingga elemen terakhir array.
- Pada setiap iterasi, elemen `key` diambil, dan sub-array sebelumnya yang sudah diurutkan digeser satu posisi ke kanan untuk membuat ruang bagi `key`.
- Elemen `key` kemudian ditempatkan pada posisi yang benar dalam sub-array.