

TUGAS PRAKTIKUM STRUKTUR DATA

Jobsheet 8 : Bubble Sort and Insertion Sort

Dosen Pengampu :

Randi Proska Sandra, M.Sc



Oleh :

Manja Fani Oktavia

22343056

PROGRAM STUDI INFORMATIKA

JURUSAN TEKNIK ELEKTRONIKA

FAKULTAS TEKNIK

UNIVERSITAS NEGERI PADANG

2023

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

Bubble Sort

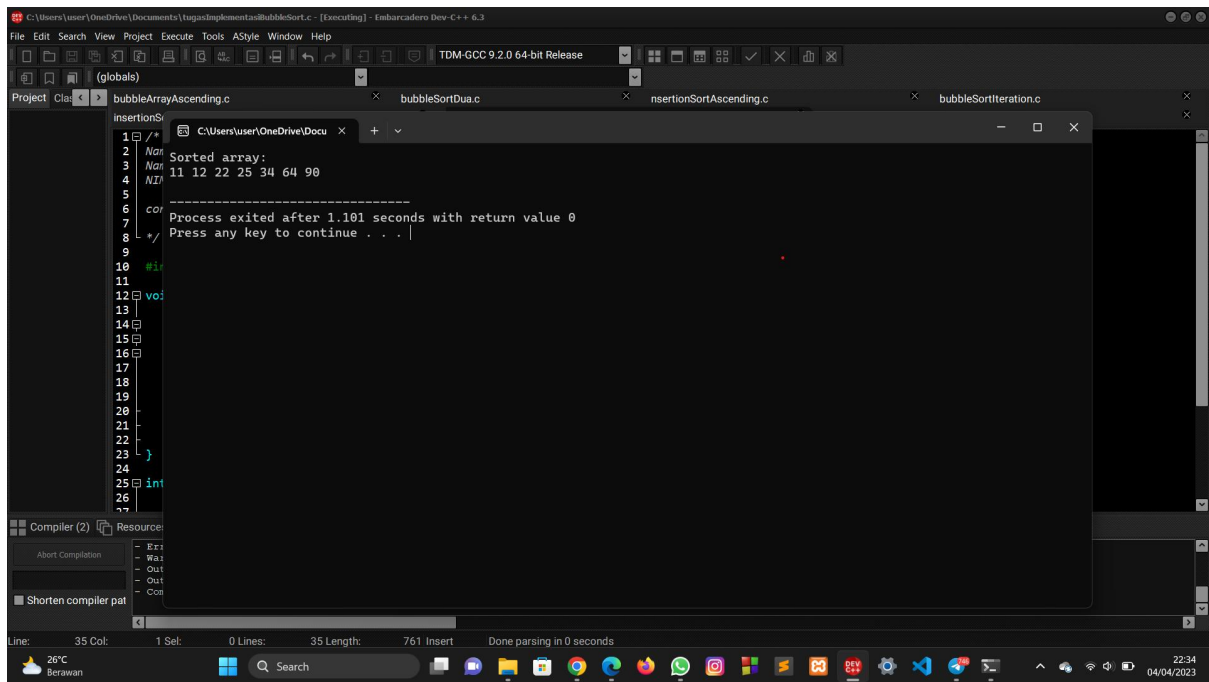
Source code :

```
/*  
Nama File : tugasImplementasiBubbleSort.c  
Nama      : Manja Fani Oktavia  
NIM       : 22343056
```

contoh aplikasi : sorting bilangan dalam sebuah array .

```
*/  
  
#include <stdio.h>  
  
void bubbleSort(int arr[], int n) {  
    int i, j, temp;  
    for (i = 0; i < n-1; i++) {  
        for (j = 0; j < n-i-1; j++) {  
            if (arr[j] > arr[j+1]) {  
                temp = arr[j];  
                arr[j] = arr[j+1];  
                arr[j+1] = temp;  
            }  
        }  
    }  
}  
  
int main() {  
    int arr[] = {64, 34, 25, 12, 22, 11, 90};  
    int n = sizeof(arr)/sizeof(arr[0]);  
    bubbleSort(arr, n);  
    printf("Sorted array: \n");  
    for (int i=0; i < n; i++)  
        printf("%d ", arr[i]);  
    printf("\n");  
    return 0;  
}
```

Hasil :



Penjelasan :

Pada bubble sort, elemen yang lebih besar akan didorong ke arah ujung array dengan melakukan pertukaran dengan elemen selanjutnya yang lebih kecil, sehingga nilai maksimum dari array bergerak ke ujung kanan. Pada setiap iterasi, nilai yang lebih kecil ditemukan dan disusun pada posisi yang lebih awal. Karena algoritma ini membandingkan setiap elemen dalam array, kompleksitas waktu rata-rata adalah $O(n^2)$.

Insertion Sort

Source code :

```

/*
Nama File : tugasImplementasiInsertionSort.c
Nama      : Manja Fani Oktavia
NIM       : 22343056

```

contoh aplikasi : sorting bilangan dalam sebuah array .

```

*/

```

```

#include <stdio.h>

```

```

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

```

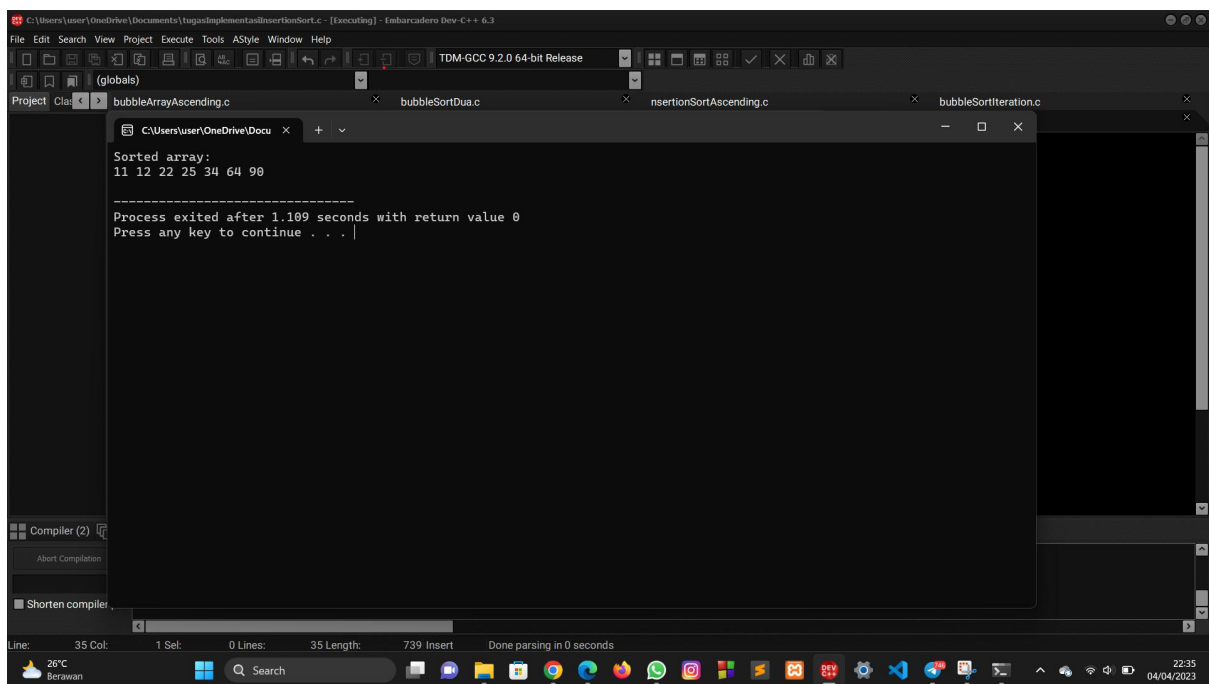
```

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

int main() {
    int arr[] = {64, 34, 25, 12, 22, 11, 90};
    int n = sizeof(arr)/sizeof(arr[0]);
    insertionSort(arr, n);
    printf("Sorted array: \n");
    for (int i=0; i < n; i++)
        printf("%d ", arr[i]);
    printf("\n");
    return 0;
}

```

Hasil :



```

C:\Users\user\OneDrive\Documents\tugasImplementasiInsertionSort.c - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Class bubbleArrayAscending.c bubbleSortDua.c nsertionSortAscending.c bubbleSortIteration.c
C:\Users\user\OneDrive\Docu x + -
Sorted array:
11 12 22 25 34 64 90

-----
Process exited after 1.109 seconds with return value 0
Press any key to continue . . .
Compiler (2)
Abort Compilation
Shorten compiler
Line: 35 Col: 1 Sel: 0 Lines: 35 Length: 739 Insert Done parsing in 0 seconds
26°C Berawan 22:35 04/04/2023

```

Penjelasan :

Pada insertion sort, elemen array dimulai dari indeks ke-1 dipindahkan ke tempat yang sesuai dalam bagian terurut dari array dengan membandingkan elemen yang lebih kecil dari indeks saat ini dan memindahkan elemen ke kanan untuk memberikan ruang pada elemen baru. Pada setiap iterasi, sebuah nilai kunci dipilih dan disisipkan ke dalam bagian terurut dari array pada tempat yang sesuai. Karena setiap elemen diurutkan satu per satu, kompleksitas waktu rata-rata adalah $O(n^2)$.