

**LAPORAN INDIVIDU
BUBLE SORT**



**Muhammad Dhafa Jawadil Ubaid
21091397058
D4 Manajemen Informatika**

1. Laporan kodingan, screenshot kode tipe sorting yang anda buat, beri penjelasan, dan bukti berupa screenshot hasil run kodingan

Codingan Buble Sort

```
#include <iostream>
#include <conio.h>
using namespace std;
main()
{
    int data [100];
    int i, j, k, tmp;
    cout<<"Pengurutan Bilangan Buble Sort \n\n";
    cout<<"Masukkan Jumlah Bilangan : ";cin>>k;
    for(i=0; i<k; i++)
    {
        cout<<"Masukkan Angka Ke "<<(i+1)<<" : ";
        cin>>data[i];
    }
    cout<<"\nAngka Sebelum Diurutkan : "<<endl;
    for(i=0; i<k; i++)
    {
        cout<<data[i]<<" ";
    }
    cout<<endl;
    for(i=0; i<k; i++)
    {
        for(j=i+1; j<k; j++)
        {
            if(data[i]>data[j])
            {
                tmp=data[i];
                data[i]=data[j];
                data[j]=tmp;
            }
        }
    }
}
```

```

    }
}

cout<<"\nAngka Setelah Diurutkan : "<<endl;
for(i=0; i<k; i++)
{
    {
        cout<<data[i]<<" ";
    }
}
getch();
}

```

The screenshot shows the Dev-C++ IDE with a project named "Buble Sort". The main window displays the source code for "buble sort.cpp". The code implements a bubble sort algorithm. It starts by including `<iostream>` and `<conio.h>`, and using the `std` namespace. In the `main` function, it declares an array `data` of size 100, and variables `i`, `j`, `k`, and `tmp`. It prompts the user to enter the number of elements (`k`) and then enters the data into the array. It prints the array before sorting, then performs the bubble sort using nested loops. The outer loop runs from `i=0` to `i<k`, and the inner loop runs from `j=i+1` to `j<k`. If `data[i] > data[j]`, it swaps the elements using a temporary variable `tmp`. Finally, it prints the sorted array and waits for a key press with `getch()`.

The bottom panel shows the "Compilation results..." window. It reports 0 errors and 0 warnings. The output filename is `E:\KULIAH\Semester 2\Struktur Data\praktek\Buble Sort.exe`, the output size is 1.8332633972168 MiB, and the compilation time is 10.92s.

```

1  #include <iostream>
2  #include <conio.h>
3  using namespace std;
4  main()
5  {
6      int data [100];
7      int i, j, k, tmp;
8      cout<<"Pengurutan Bilangan Buble Sort \n\n";
9      cout<<"Masukkan Jumlah Bilangan : ";cin>>k;
10     for(i=0; i<k; i++)
11     {
12         cout<<"Masukkan Angka Ke "<<(i+1)<<" : ";
13         cin>>data[i];
14     }
15     cout<<"\nAngka Sebelum Diurutkan : "<<endl;
16     for(i=0; i<k; i++)
17     {
18         cout<<data[i]<<" ";
19     }
20     cout<<endl;
21     for(i=0; i<k; i++)
22     {
23         for(j=i+1; j<k; j++)
24         {
25             if(data[i]>data[j])
26             {
27                 tmp=data[i];

```

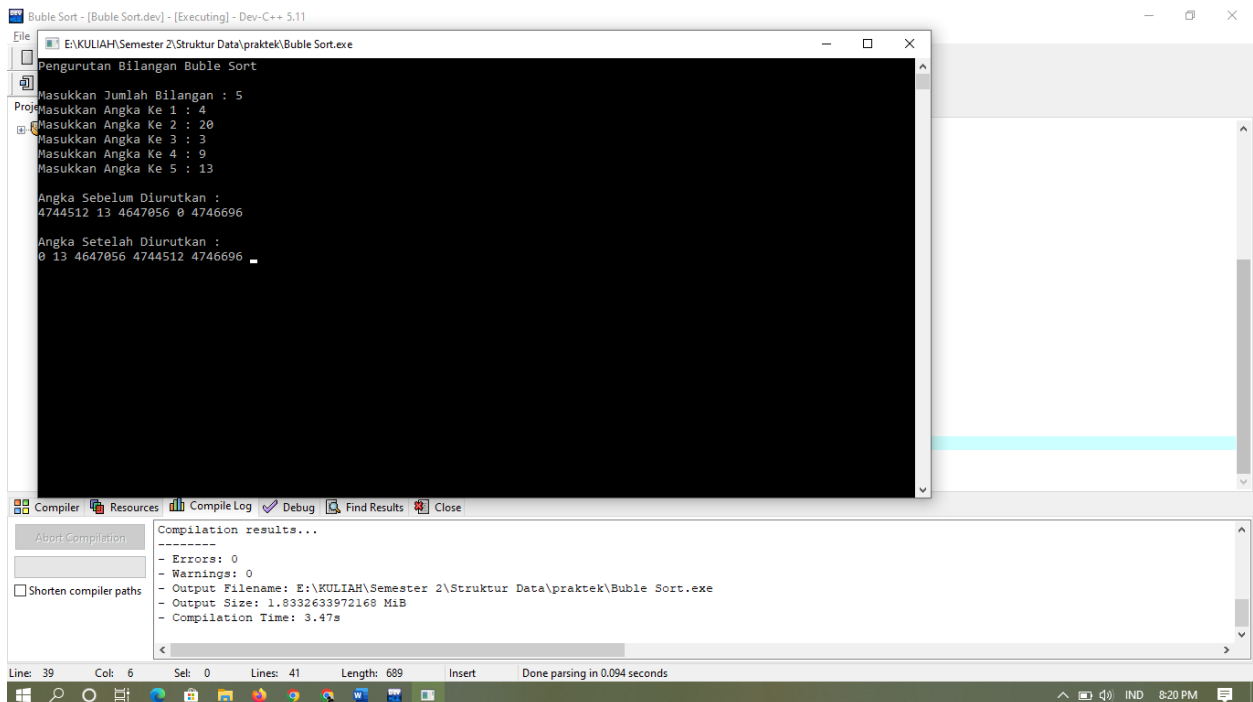
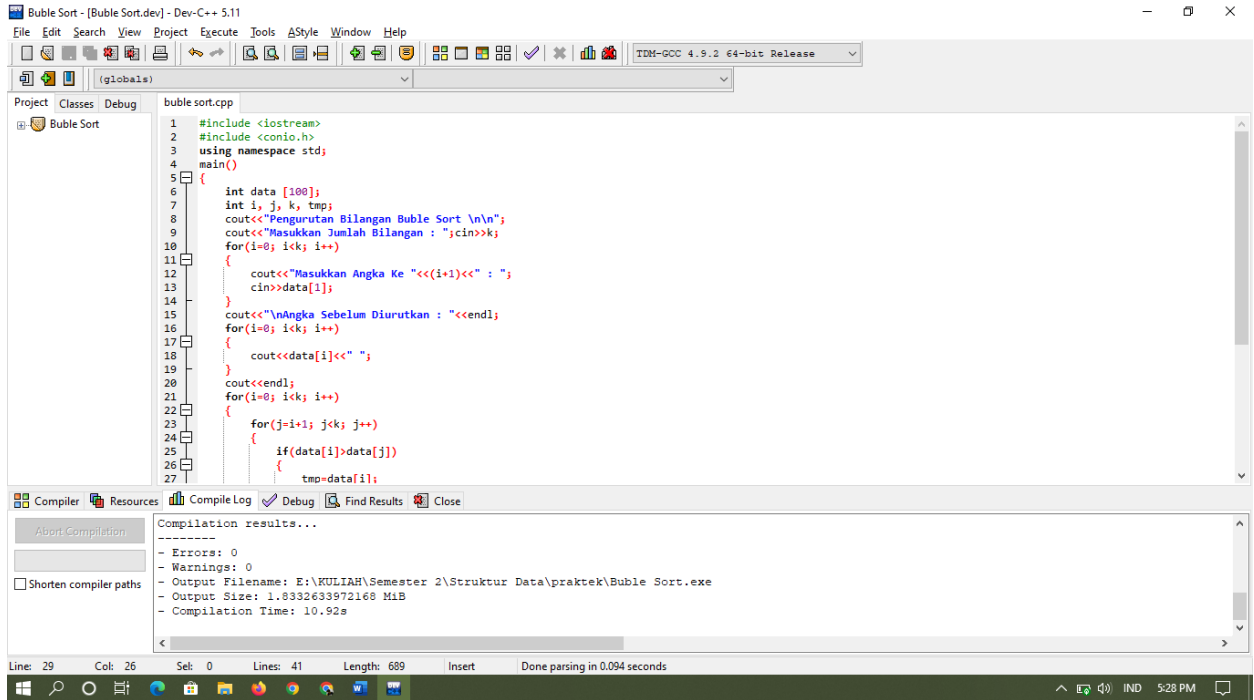
Compilation results...

```

-----
- Errors: 0
- Warnings: 0
- Output Filename: E:\KULIAH\Semester 2\Struktur Data\praktek\Buble Sort.exe
- Output Size: 1.8332633972168 MiB
- Compilation Time: 10.92s

```

Line: 29 Col: 26 Sel: 0 Lines: 41 Length: 689 Insert Done parsing in 0.094 seconds



Penjelasan

Bubble sort adalah sebuah teknik pengurutan data dengan cara menukar dua data yang bersebelahan jika urutan dari data tersebut salah. Algoritma ini dapat mengurutkan data dari besar ke kecil (Ascending) dan kecil ke besar (Descending). Algoritma ini tidak cocok untuk set data dengan jumlah besar karena kompleksitas dari algoritma ini adalah $O(n^2)$ di mana n adalah jumlah item.

Banyak Perbandingan = $(\text{jumlah data} - 1) \times (\text{jumlah data} - 1)$

2. Hitung jenis Big O (worst case scenario untuk time complexity) nya, jelaskan kenapa kompleksitasnya adalah yang anda temukan.

- Hitung Big O saat $n=1$
- Hitung Big O saat $n=5$
- Hitung Big O saat $n=10$
- Jelaskan menurut anda Big O nya termasuk jenis yang mana dan jelaskan kenapa

3. Jelaskan kelebihan dan kekurangan sorting yang kalian buat dibandingkan yang dibuat oleh teman kalian.

Kelebihan :

- a) proses perhitungan dengan metode yang paling sederhana,
- b) mudah dipahami, dan
- c) tahapan dalam pengurutan data sangat sederhana

Kekurangan :

- a) proses perhitungan menggunakan metode pengurutan yang paling tidak efisien meskipun sederhana,
- b) proses perhitungan akan lambat atau lama apabila data dengan jumlah besar. Jadi untuk proses pengurutan data secara tunggal
- c) jumlah pengulangan akan tetap sama sampai data yang terakhir, meskipun sebagian data ada telah diurut