Nama :Nadia Alifiani Raissa Pansera

Nim :21091397014

LAPORAN INDIVIDU

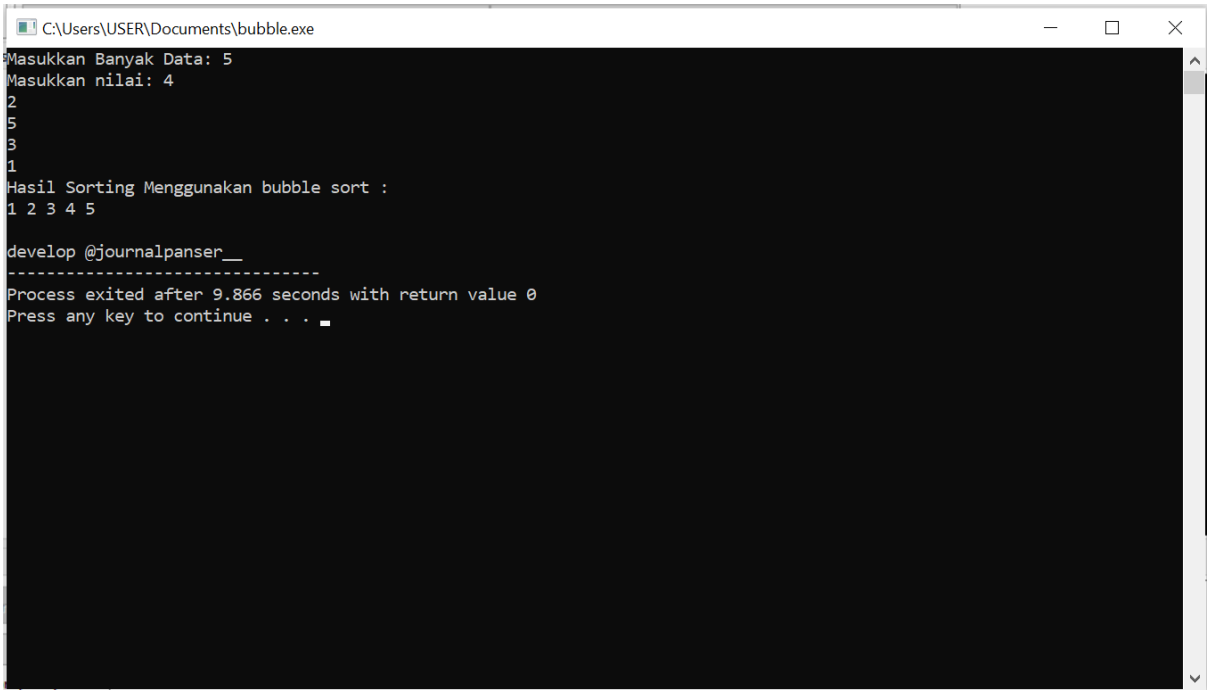
Kode tipe :

bubble.cpp

```cpp
#include <iostream>
using namespace std;

//a function to implement bubble sort
void bubbleSort(int arr[], int n){
  int i, j, tmp;
  for (i = 0; i < n; i++){
    //last i elements are already in place
    for (j = 0; j < n - i - 1; j++){
      if (arr[j] > arr[j + 1]){
        tmp = arr[j];
        arr[j] = arr[j + 1];
        arr[j + 1] = tmp;
      }
    }
  }
}
int main(){
    //declaration
  int array[100], n, i, j;
  //input the element
  cout << "Masukkan Banyak Data: ";
  cin >> n;
  cout << "Masukkan nilai: ";
  //function to print an array
  for (i = 0; i < n; i++){
    cin >> array[i];
  }
  bubbleSort(array, n);
  //result
  cout << "Hasil Sorting Menggunakan bubble sort :\n";
  for (i = 0; i < n; i++){
    cout << array[i] << " ";
  }
  cout << "\n";
  cout << "\ndevelop @journalpanser__";
}
```

Hasil Run ;

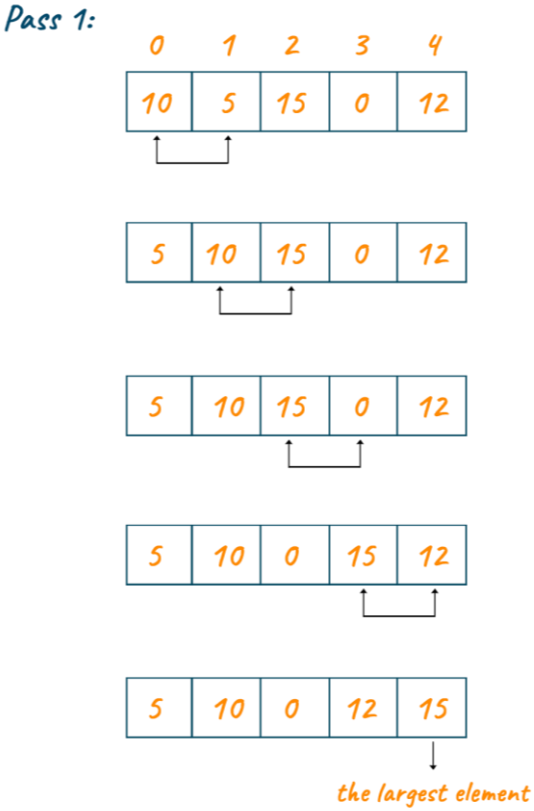


Kompleksitas waktu algoritma bubble sort adalah O(n) untuk skenario best-case ketika array benar-benar diurutkan. Mempertimbangkan kasus rata-rata dan skenario terburuk, kompleksitas waktu bubble sort adalah O (n ^ 2) di mana n adalah jumlah total elemen dalam array. Itu karena kita harus menggunakan dua loop yang melintasi seluruh array untuk menyortir elemen.

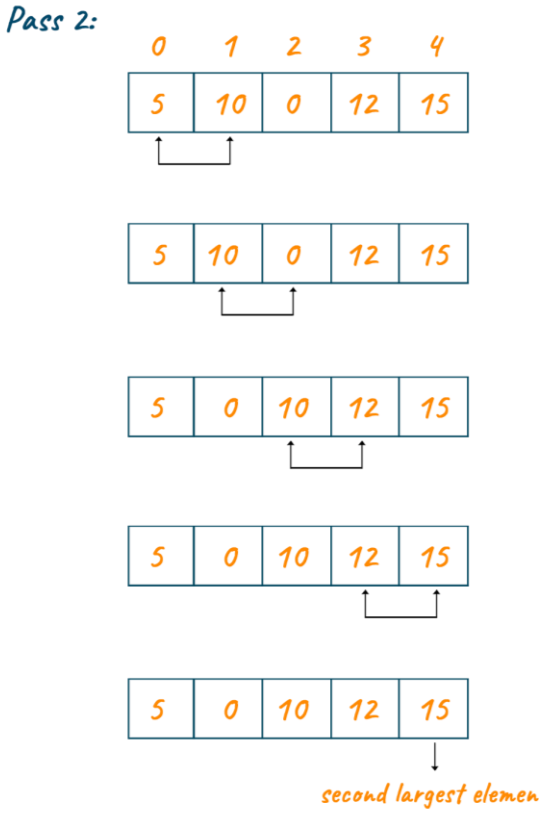
Penjelasan :

Kita mulai dengan elemen '0th' di loop pertama, lalu kita mulai dengan elemen yang berdekatan di loop kedua. Kami membandingkan masing-masing komponen tetangga dalam tubuh loop bagian dalam dan menukarnya jika tidak teratur. Elemen terberat menggelembung di akhir setiap iterasi loop luar.

elemen terbesar gelembung hingga yang terakhir dengan setiap pass, menyortir daftar. Setiap elemen dibandingkan dengan elemen yang berdekatan dan beralih satu sama lain jika tidak teratur.



Jika array akan diurutkan dalam urutan menaik pada akhir pass pertama, elemen terbesar ditempatkan di akhir daftar. Elemen terbesar kedua ditempatkan pada posisi terakhir kedua dalam daftar untuk pass kedua, dan seterusnya.



Akhirnya, kita akan mengurutkan seluruh daftar ketika n-1 (di mana n adalah jumlah total entri dalam daftar) berlalu. Output akhir dari jenis gelembung akan seperti yang ditunjukkan di bawah ini.

Kelebihan & kelemahan buble sort :

1. Kelebihan :
   - Metode Sorting Termudah
   - Metode Buble Sort Mudah Dipahami Algoritmanya
   – Mudah Untuk Diubah Menjadi Kode.
   – Definisi Terurut Terdapat Dengan Jelas Dalam Algoritma.
   – Cocok Untuk Pengurutan Data Dengan Elemen Kecil Telah Terurut
2. Kekurangan :
   - Tidak efisien
   - Pada Saat Mengurutkan Data Yang Sangat Besar Akan Mengalami Kelambatan.
   - Jumlah Pengulangan Akan Tetap Sama Jumlahnya Walaupun Data Sesungguhnya Sudah Cukup Terurut.

-