

NAMA : RAHMADITYA PUTRI LAILATUL ISMI
KELAS : D4 MANAJEMEN INFORMATIKA B

LAPORAN INDIVIDU CODING "BUBBLE SORT"

✚ Bubble Sort merupakan metode sorting dengan membandingkan elemen-elemen yang ada di dalam tabel. Metode ini mendasar pada pertukaran dua buah elemen untuk mencapai keadaan urut yang diinginkan.

✚ Algoritma Bubble Sort :

- Membandingkan data ke- i dengan data ke- $(i+1)$ (tepat bersebelahan). Jika tidak sesuai maka tukar (data ke- i = data ke- $(i+1)$ dan data ke- $(i+1)$ = data ke- i). Apa maksudnya tidak sesuai? Jika kita menginginkan algoritme menghasilkan data dengan urutan ascending (A-Z) kondisi tidak sesuai adalah data ke- $i >$ data ke- $i+1$, dan sebaliknya untuk urutan descending (A-Z).
- Membandingkan data ke- $(i+1)$ dengan data ke- $(i+2)$. Kita melakukan perbandingan ini sampai data terakhir. Contoh: 1 dgn 2; 2 dgn 3; 3 dgn 4; 4 dgn 5 ... ; $n-1$ dgn n .
- Selesai satu iterasi, adalah jika kita sudah selesai membandingkan antara $(n-1)$ dgn n . Setelah selesai satu iterasi kita lanjutkan lagi iterasi berikutnya sesuai dengan aturan ke1. mulai dari data ke-1 dgn data ke-2, dan seterusnya.
- Proses akan berhenti jika tidak ada pertukaran dalam satu iterasi

✚ Kelebihan Bubble Sort :

- Metode Bubble Sort merupakan yang paling simple ➤ Metode Bubble Sort mudah di pahami algoritmanya ➤ Mudah untuk diubah menjadi kode.
- Definisi terurut terdapat dengan jelas dalam algoritma.
- Cocok untuk pengurutan data dengan elemen kecil telah terurut.

Kekurangan Bubble Sort :

- ✚
1. Metode Bubble Sort merupakan metode pengurutan yang paling tidak efisien
 2. Pada saat mengurutkan data yang sangat besar akan mengalami kelambatan luar biasa



CODE BUBBLE SORT

```
D:\pindahan C:\Documents\Manajemen Informatika\SEMESTER 2\STRUKTUR DATA\036_Rahmaditya P_Bubble Sort.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools Style Window Help
gdbgui
036_Rahmaditya P_Bubble Sort.cpp

#include <iostream>

using namespace std; //deklarasi namespace agar memudahkan penggunaan fungsi cin dan cout tidak lagi menambati diawal beta cout & cin dengan penulisan std.

void bubbleSort(int arr[], int n) //deklarasi fungsi void bubbleSort dengan menerima inputan arr dan n yang ber tipe integer
{
    int i, j, temp; //deklarasi variable i dan j untuk perulangan sedangkan temp untuk variabel pertukaran sementara
    for (i = 0; i < n; i++) //perulangan sebanyak n dan perulangan sebanyak n dan perulangan sebanyak n dikurangi 1 dan dikurangi satu
    {
        for (j = i + 1; j < n - i; j++) //cek kondisi dengan menggunakan if jika nilai variable arr pada index ke j lebih besar dari nilai variable arr pada index ke j-1 maka melakukan pertukaran
        {
            if (arr[j] > arr[j - 1])
            {
                temp = arr[j];
                arr[j] = arr[j - 1];
                arr[j - 1] = temp;
            }
        }
    }
}

int main()
{
    int array[10], data_element, i, j; //mendeklarasi variable array, n, i dan j ber tipe integer
    cout << "Masukkan banyak elemen: "; //meminta input nilai dari user untuk variable n yang digunakan sebagai jumlah data
    cin >> data_element;
    cout << "Masukkan nilai: "; //meminta input nilai dari user untuk variable array dari setiap data
    for (i = 0; i < data_element; i++)
    {
        cin >> array[i];
    }
    bubbleSort(array, data_element); //memanggil fungsi bubbleSort dengan memasukkan variable array dan n
    cout << "Hasil array yang sudah di sorting:\n"; //menampilkan hasil pengurutan bubbleSort
    for (i = 0; i < data_element; i++)
    {
        cout << " " << array[i] << " ";
    }
    cout << "\n";
    return 0;
}
```



OUTPUT BUBBLE SORT

```
D:\pindahan C:\Documents\Manajemen Informatika\SEMESTER 2\STRUKTUR DATA\036_Rahmaditya P_Bubble Sort.exe
Masukkan banyak elemen: 6
Masukkan nilai:
9
4
7
3
8
2
Hasil Array yang sudah di sorting:
[2] [3] [4] [7] [8] [9]
-----
Process exited after 133.5 seconds with return value 0
Press any key to continue . . .
```

🚦 Contoh proses sorting bubble sort

A	21	25	7	8	11	13
	1	2	3	4	5	6

Langkah 1 : dimulai dengan mengakses indeks pertama dari array dan membandingkannya dengan indeks setelahnya (indeks kedua).

A	21	25	7	8	11	13
	1	2	3	4	5	6

Karena elemen pertama tidak lebih besar dari elemen kedua, maka tidak akan dilakukan pertukaran.

Kemudian dimulai kembali dengan membandingkan indeks kedua dengan indeks setelahnya (indeks ketiga).

A	21	25	7	8	11	13
	1	2	3	4	5	6

Karena elemen kedua lebih besar dari elemen ketiga, maka dilakukan pertukaran.

A	21	7	25	8	11	13
	1	2	3	4	5	6

Selanjutnya, dilakukan dengan aturan yang sama. Apabila nilai di suatu indeks lebih besar dari nilai di indeks setelahnya, maka pertukaran nilai akan dilakukan.

A

21	7	25	8	11	13
1	2	3	4	5	6

A

21	7	8	25	11	13
1	2	3	4	5	6

A

21	7	8	11	25	13
1	2	3	4	5	6

Setelah dilakukan pembandingan sebanyak 5 kali, akhirnya didapatkan hasil pengurutan langkah pertama sebagai berikut.

A

21	7	8	11	13	25
1	2	3	4	5	6

Langkah 2 : didapatkan hasil pembandingan sebagai berikut :

A

21	7	8	11	13	25
1	2	3	4	5	6

A

7	21	8	11	13	25
1	2	3	4	5	6

A

7	8	21	11	13	25
1	2	3	4	5	6

A

7	8	11	21	13	25
1	2	3	4	5	6

Maka, setelah dilakukan pembandingan sebanyak 4 kali, akhirnya didapatkan hasil pengurutan langkah kedua sebagai berikut.

A

7	8	11	13	21	25
1	2	3	4	5	6

Langkah 3 : didapatkan hasil pembandingan sebagai berikut :

A

7	8	11	13	21	25
1	2	3	4	5	6

A

7	8	11	13	21	25
1	2	3	4	5	6

A

7	8	11	13	21	25
1	2	3	4	5	6

Maka, setelah dilakukan pembandingan sebanyak 3 kali, akhirnya didapatkan hasil pengurutan langkah ketiga sebagai berikut.

A

7	8	11	13	21	25
1	2	3	4	5	6

Langkah 4 : didapatkan hasil pembandingan sebagai berikut :

A

7	8	11	13	21	25
1	2	3	4	5	6

A

7	8	11	13	21	25
1	2	3	4	5	6

Maka, setelah dilakukan pembandingan sebanyak 2 kali, akhirnya didapatkan hasil pengurutan langkah keempat sebagai berikut.

A

7	8	11	13	21	25
1	2	3	4	5	6

Langkah 5 : didapatkan hasil pembandingan sebagai berikut :

A

7	8	11	13	21	25
1	2	3	4	5	6

Setelah dilakukan pembandingan sebanyak 1 kali, didapatkan hasil pengurutan langkah kelima sebagai berikut.

A

7	8	11	13	21	25
1	2	3	4	5	6

Maka, pada akhirnya array dapat dipastikan terurut setelah langkah kelima dikerjakan.

A

7	8	11	13	21	25
1	2	3	4	5	6