

Nama : Syarifah Saskia Aulia

Nim : 21091397012

Kelas : B

LAPORAN INDIVIDU

- Kode tipe Insertion Sort:

```
[*] Algoritma Insertion Sort.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int jumlah_array;
7
8      cout<<"masukan banyak array:";
9      cin>>jumlah_array;
10
11     int angka[jumlah_array];
12
13     for(int i=0; i<jumlah_array; i++){
14         cout<<"masukan angka ke "<<i<<" :";
15         cin>>angka[i];
16         cout<<endl;
17     } // perulangan untuk menampilkan angka sesuai data angka pada baris 11
18
19     for(int i=1; i<jumlah_array; i++){
20         int key = angka[i];
21         int j = i-1;
22         while(j>=0 && angka[j] > key){
23             angka[j+1] = angka[j];
24             j--;
25         }
26         angka[j+1] = key;
27         cout<<"proses sorting"<<endl;
28         for(int m=0; m<jumlah_array; m++){
29             cout<<angka[m]<<" ";
30         }
31         cout<<endl;
32     }
33
34     cout<<"hasil akhir"<<endl;
35     for(int m=0; m<jumlah_array; m++){
36         cout<<angka[m]<<" ";
37     }
38 }
```

- Penjelasan :
Terdapat array dengan 5 elemen dibawah ini :

12	4	8	10	7
0	1	2	3	5

Selanjutnya data pada index 0 dan 1 akan dipindahkan ke sorted list. Setelahnya, data pada index 0 dan 1 di sorted list akan dibandingkan untuk mencari index yang memiliki nilai terkecil. Dari data di atas, terlihat bahwa nilai pada index 1 lebih kecil dari nilai index 0. Maka pada sorted list, terjadi perubahan posisi. Hasilnya dapat dilihat pada gambar di bawah ini :

Sorted List

4	12
0	1

Unsorted List

8	10	7
2	3	4

Langkah 2. Data pada index 2 dipindahkan ke sorted list. Lalu, data tersebut dibandingkan dengan data-data pada index 0 dan 1. Pada langkah kali ini, hasilnya data pada index 2 memiliki nilai lebih kecil ketimbang data di index 1. Hasilnya, data yang baru masuk sorted list tersebut diposisikan ke index 1. Hasil komparasinya akan seperti gambar berikut:

Sorted List

4	8	12
0	1	2

Unsorted List

10	7
3	4

Langkah 3. Data pada index 3 masuk ke sorted list dan dibandingkan dengan data-data yang sudah ada di sorted list. Karena nilai data pada index 3 lebih kecil daripada nilai data di index 2, maka data pada index 3 diposisikan ke index 2, dan data pada index 2 bergerak mundur. Jadinya akan seperti berikut:

Sorted List

4	8	10	12
----------	----------	-----------	-----------

0 1 2 3

Unsorted List

7
4

Langkah 4. Langkah terakhir adalah memasukkan data pada index 4 di unsorted list ke dalam sorted list. Setelah dibandingkan, nilai pada data index 4 lebih kecil ketimbang nilai data pada index 1 sampai 3. Hasilnya, data index 4 akan diposisikan ke index 1 dan data setelahnya akan bergerak mundur. Maka, hasilnya akan seperti gambar di bawah ini:

Sorted List

4	7	8	10	12
1	2	3	4	5

Dengan hasil tersebut, maka proses pengurutan dengan metode insertion sort sudah selesai.

- SS hasil Run:

```
C:\Users\User\Documents\Insert1.exe

masukan banyak array:5
masukan angka ke 0 :12

masukan angka ke 1 :4

masukan angka ke 2 :8

masukan angka ke 3 :10

masukan angka ke 4 :7

proses sorting
4 12 8 10 7
proses sorting
4 8 12 10 7
proses sorting
4 8 10 12 7
proses sorting
4 7 8 10 12
hasil akhir
4 7 8 10 12
-----
Process exited after 19.67 seconds with return value 0
Press any key to continue . . .
```

- Kelebihan dan Kekurangan Algoritma Insertion Sort :
 - **Kelebihan :**
 1. Insertion sort adalah salah satu algoritma sorting tercepat daripada algoritma yang lainnya.
 2. Sangat cocok untuk melakukan sorting dengan jumlah data yang besar.
 3. Karena list langsung di sort di tempat, maka tidak memerlukan memory tambahan untuk menjalankannya.
 - **Kekurangan :**
 1. Jika data sebelumnya sudah di ada yang ter sorting maka algoritma ini kurang efisien, karena ia akan mengecek seluruh data di dalam list.
 2. Lebih cocok mensorting bilangan yang bentuknya bulat.