Nama : Nur Haslinda

Nim : 21091397035

Kelas : 2021 A

Program C++ Insertion sort

```
1  //NUR HASLINDA_21091397035
2
3  #include<iostream>
4  using namespace std;
5
6  int main(){
7
8      //Inisialisasi Variabel dengan tipe data integer
9      int looping1, looping2, batas_elemen, temp, input_elemen[50];
10
11
12     cout<<"\t\tSHORTING MENGGUNAKAN INSERTION SORT\n";
13
14
15     cout<<"\nInput Batas Elemen Array   : ";    //Input Batas array menggunakan angka
16     cin>>batas_elemen;
17     cout<<"\nInput List Array   : \n";          //Input elemen array menggunakan angka
18
19
20     //looping untuk memasukkan list array
21     for(looping1=0; looping1<batas_elemen; looping1++)
22     {
23         cout << "Elemen ke-"<<looping1<< ": ";
24
25         //perintah memasukkan elemen erray
26         cin>>input_elemen[looping1];
27     }
28
29     //looping untuk proses sorting
30     for(looping1=1; looping1<=batas_elemen-1; looping1++)
31     {
32         //Variabel untuk menyimpan array sementara
33         temp=input_elemen[looping1];
34         looping2=looping1-1;
35
```

```
35
36          while((temp<input_elemen[looping2])&&(looping2>=0))
37          {
38              //perintah memindahkan elemen array kedepan sesuai urutan
39              input_elemen[looping2+1]=input_elemen[looping2];
40              looping2=looping2-1;
41          }
42
43          //Perintah untuk mengurutkan elemen array dari yang terkecil
44          input_elemen[looping2+1]=temp;
45      }
46
47      cout<<"\nHASIL SETELAH SORTING\n";
48
49      //looping untuk menampilkan hasil sorting
50      for(looping1=0; looping1<batas_elemen; looping1++)
51      {
52          //Perintah menampilkan hasil sorting
53          cout<<"[" <<input_elemen[looping1]<< "] ";
54      }
55
56      return 0;
57  }
```

Output Program

```
               SHORTING MENGGUNAKAN INSERTION SORT

Input Batas Elemen Array        : 5

Input List Array        :
Elemen ke-0: 3
Elemen ke-1: 4
Elemen ke-2: 5
Elemen ke-3: 1
Elemen ke-4: 2

HASIL SETELAH SORTING
[1] [2] [3] [4] [5]
--------------------------------
Process exited after 8.367 seconds with return value 0
Press any key to continue . . .
```

Pengertian Insertion sort

Insertion sort merupakan teknik pengurutan dengan cara membandingkan dan mengurutkan dua data pertama pada array, kemudian membandingkan data array berikutnya apakah sudah berada pada tempat semestinya.

Cara kerja algoritma insertion sort

Mengambil elemen list satu per-satu dan memasukkannya di posisi yang benar. Pengurutan insertion sort merupakan pengurutan data dengan membandingkan dua elemen data pertama, kemudian membandingkan elemen-elemen data yang sudah diurutkan, perbandingan akan terus diulang hingga tidak ada data yang tersisa. Untuk mengehmat memori, implementasi insertion sort menggunakan pengurutan di tempat yang membandingkan elemen saat itu dengan elemen sebelumnya yang sudah diurut, lalu menukarnya terus sampai posisinya tepat.

Contoh proses Shorting insertion short

2	1	6	4	5

Pada kolom ini merupakan elemen list array yang masih belum tersorting yaitu [2, 1, 6, 4, 5].

2	1	6	4	5

Elemen list kedua bertukar dengan elemen list pertama karena elemen list kedua kebih kecil dari elemen list pertama.

1	2	6	4	5

Elemen list ketiga tidak bertukar dengan elemen list kedua dan list pertama karena elemen list ke tiga lebih besar dari elemen list kedua dan pertama. Elemen list keempat bertukar dengan elemen list ketiga karena elemen list ketiga lebih besar dari elemen list pertama.

1	2	4	6	5

Elemen list kelima bertukar dengan elemen list keempat karena elemen list kelima lebih kecil dari elemen list keempat. Proses tukar berhenti di elemen list keempat karena elemen list ketiga, dua, dan satu lebih kecil dari elemen list keempat.

1	2	4	5	6

Proses shorting berhenti karena semua elemen sudah tershorting sesuai urutan.

Kompleksitas Insetion Sort

Pada algoritma insertion sort terdiri dari 2 kalangan bersarang. Dimana terjadi N-1 elemen (dengan N adalah banyak elemen struktur data), dengan masing-masing elemen terjadi *i* kali operasi perbandingan. *i* tersebut bernilai 1 untuk elemen pertama, bernilai 2 untuk elemen kedua, begitu seterusnya hingga ke elemen N-1. Oleh karena itu kompleksitas algoritma insertion sort memiiki kompleksitas **waktu terbaik (*best case*)** adalah **O(n)**, kompleksitas **waktu terburuk (*worst case*)** adalah **$O(n^2)$**, dan kompleksitas waktu **rata-rata (*Average*)** adalah **$O(n^2)$**.

$T(n) = 1+2+..+n-1= \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}=O(n^2)$

Big O N = 1

Big O N = 1

Big O $N^2 = 1^2$

Big O = 1

Big O N=5

Big O N = 5

Big O $N^2 = 5^2$

Big O = 25

Big O N=10

Big O N = 10

Big O $N^2 = 10^2$

Big O = 100

Kelebihan algoritma insertion sort

1. Salah satu algoritma Sorting tercepat
2. Implementasi yang sederhana.
3. Karena list langsung di sort di tempat, maka tidak memerlukan memory tambahan untuk menjalankannya.
4. Paling efisien untuk proses sorting data yang berukuran kecil.
5. Algoritma stabil dengan proses sorting tercepat pada jumlah elemen yang sedikit.

Kelemahan algoritma insertion sort

1. Kurang efisien jika digunakan untuk data yang sebelumnya sudah di ada yang tersorting, karena program akan mengecek seluruh data di dalam list.
2. Lebih cocok mensorting bilangan yang bentuknya bulat.
3. Tidak praktis untuk elemen yang berjumlah banyak.
4. Tidak disarankan untuk menangani struktur data dengan jumlah elemen lebih dari 2000 elemen.