

Nama : Bagas Syafiq Aero Pradana

NIM : 21091397064/kelas B

LAPORAN INDIVIDU INSERTION SORT

➤ Script Insertion Sort

```
dev C:\Users\asus\Downloads\Struktur data\yyy\INSERTION SORT.cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug INSERTION SORT.cpp
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     int i,n,j,temp;
7     char A[100];
8     cout << "Insetion ascending\n";
9     cout << "Jumlah data : ";
10    cin >> n;
11    cout << endl;
12    for (i = 0; i < n; i++){
13        cout << "Data ke - " << i+1 << " : ";
14        cin >> A[i];
15    }
16    for(i = 0; i < n; i++){
17        temp = A[i];
18        j = i-1;
19        while(A[j] > temp && j >= 0){
20            A[j+1] = A[j];
21            j--;
22        }
23        A[j+1] = temp;
24    }
25
26    cout << "Data Setelah Di Urutkan : \n";
27    for(i = 0; i < n; i++){
28
29
30        cout << A[i] << " ";
31
32    }
33
34
35
36    return 0;
37
38 }
```

OUTPUT Script Insertion Sort

```
C:\Users\asus\Downloads\Struktur data\yyyy\INSERTION SORT.exe
Insetion ascending
Jumlah data : 6

Data ke - 1 : 8
Data ke - 2 : 9
Data ke - 3 : 3
Data ke - 4 : 5
Data ke - 5 : 2
Data ke - 6 : 1
Data Setelah Di Urutkan :
1 2 3 5 8 9
-----
Process exited after 33.78 seconds with return value 0
Press any key to continue . . .
```

➤ Penjelasan Proses coding pada Insertion Sort

Algoritma insertion sort pada dasarnya memilah data yang akan urutkan menjadi 2 bagian, yang belum diurutkan dan yang sudah diurutkan. Elemen pertama diambil dari bagian array yang belum diurutkan dan kemudian diletakkan sesuai posisinya pada bagian lain dari array yang telah diurutkan. Langkah ini dilakukan secara berulang hingga tak ada lagi elemen tersisa pada bagian array yang belum diurutkan. Metode insection sort merupakan metode yang mengurutkan bilangan-bilangan yang telah terbaca, dan berikutnya secara berulang akan menyisipkan bilangan-bilangan dalam array yang belum terbaca ke sisi kiri array yang telah terurut. Kita mengambil pada bilangan yang paling kiri. Bilangan tersebut dikatakan urut terhadap dirinya sendiri karena bilangan yang di bandingkan baru 1.

Seperti penjelasan dibawah ini :

1. Input kan jumlah data pada line 4-10 yang akan disorting
2. Lalu inputkan jumlah data awal yang akan disorting pada line 16-23 sebanyak jumlah yang akan ditentukan pada inputan yang diingkan
3. Pada hasil output yang sebelum terdapat 6 jumlah input seperti dibawah ini

8	9	3	5	2	1
---	---	---	---	---	---

4. Kemudian kita mulai proses sortingan diatas ini agar bisa jadi urut sesuai urutan, jadi kita urutkan pertama kita cek angka (8) apakah lebih dari angka (9) jika benar angka (9) tetap diposisi {jawabannya benar}, lalu cek lagi untuk angka (3) apakah lebih dari angka (8) jika salah angka (3) akan bergeser kedepan angka (8) {jawabannya salah}, lalu cek lagi untuk angka selanjutnya contoh seperti diatas

5. Jadi hasilnya akan seperti dibawah ini secara berututan

1	2	3	5	8	9
---	---	---	---	---	---

- Menghitung hasil jenis BIG O
Besok katanya
- Kekurangan algoritma Insertion sort
 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
- Kelebihan algoritma insertion sort
 1. Salah satu algoritma Sorting tercepat
 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