Laporan individu

Selection sort

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

int main() {

// Variable terdiri dari jumlah_array yang berisi jumlah array yang akan diurutkan, tukar digunakan untuk mengurutkan data, dan isi adalah data yang akan diurutka
    int jumlah_array, tukar, isi[100];

// input jumlah array
    cout << "Masukkan jumlah data: ";
    cin >> jumlah_array;

// menginputkan data sebanyak jumlah array
    for (int i = 0; i < jumlah_array; i++) {
        cout << "Data ke-" << i + 1 << " = " ;
        cin >> isi[i];
        cout << endl;

    }

// proses perulangan data
    for (int i = 0; i < jumlah_array-1; i++) {
        tukar = i;

// proses menukarkan data, jika data setelahnya lebih kecil maka akan ditukar dengan data sebelumnya sampai persyaratan terpenuhi
        int temp;

        for(int j = i+1; j < jumlah_array; j++){
            if(isi[j] < isi[tukar]){
                tukar = j;
            }
        }

        temp = isi[tukar];
        isi[tukar] = isi[i];
        isi[i] = temp;
    }


// menampilkan output data secara urut
    cout << "Deret array yang sudah disortir: ";
    for(int i = 0; i < jumlah_array; i++){
        cout << "["<<isi[i] << "]";
    }

    cin.get();

    return 0;
}
```

Cara Kerja Selection sort:

1. Mencari nilai minimum(ascending) atau maksimum(descending) pada sebuah list.
2. Menukar nilai tersebut pada kolom pertama list.
3. Mengulangi langkah diatas dengan dimulai pada kolom kedua list.

Kelebihan Selection sort:

- Algoritmanya mudah diimplementasikan.
- Mudah menentukan data maksimum dan minimum.
- Kompleksitas selection sort relatif kecil.

Kelemahan Selection sort:

- Kompleksitas selection sort akan meningkat dan tidak praktis jika jumlah data mencapai 1000.
- Membutuhkan metode tambahan.
- Dibandingkan dengan insertion sort, algoritma dari selection memiliki kesamaan dalam mudahnya diimplementasikan namun memiliki performa yang lebih buruk dari insertion sort.

Big O Selection sort.

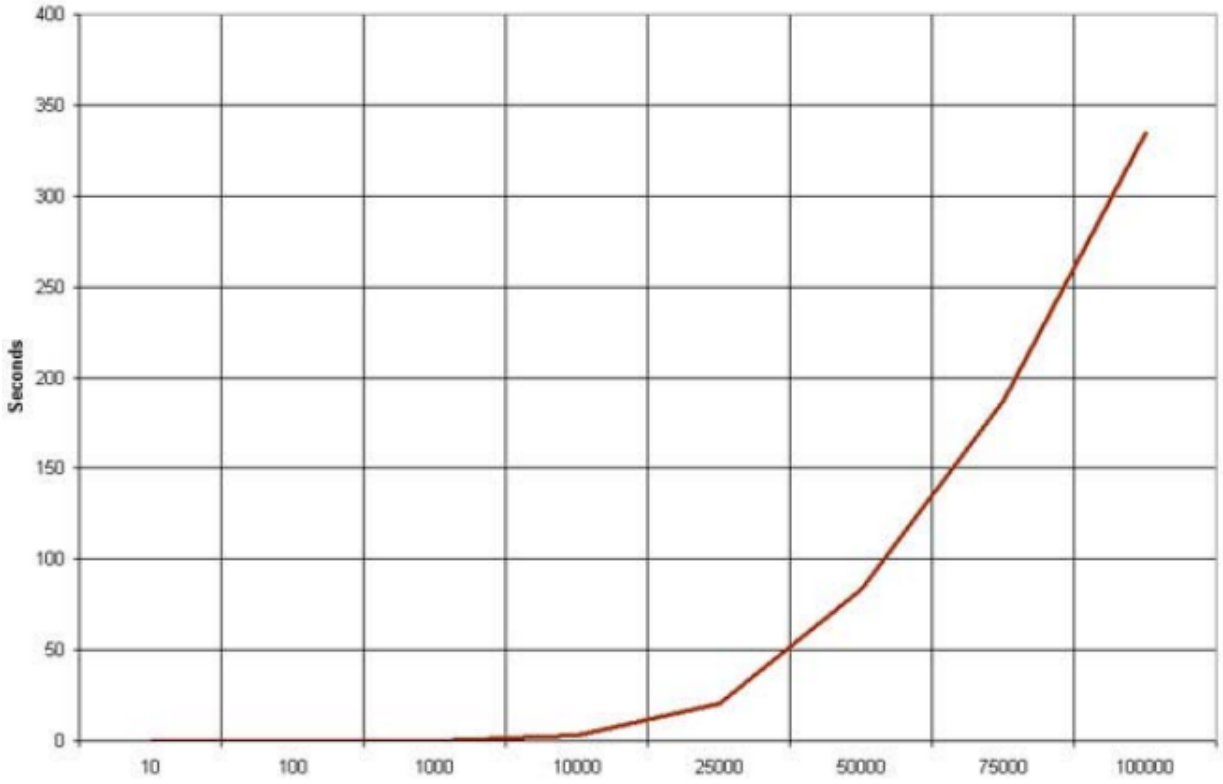
Algoritma di dalam Selection Sort terdiri dari kalang bersarang. Dimana kalang tingkat pertama (disebut pass) berlangsung N-1 kali. Di dalam kalang kedua, dicari elemen dengan nilai terkecil. Jika didapat, indeks yang didapat ditimpakan ke variabel min. Lalu dilakukan proses penukaran. Begitu seterusnya untuk setiap Pass. Pass sendiri makin berkurang hingga nilainya menjadi semakin kecil.

Pass								
0		5	1	43	27	6	18	33
1		1	5	43	27	6	18	33
2		1	5	43	27	6	18	33
3		1	5	6	27	43	18	33
4		1	5	6	18	43	27	33
5		1	5	6	18	27	43	33
6		1	5	6	18	27	33	43

Seperti yang bisa dilihat pada contoh diatas, terdapat 7 nilai yang harus diurutkan pada sort tersebut, dan untuk mengurutkan 1 pass atau nilai dibutuhkan 1 x loop atau sebanyak 7 x proses berpindah dari satu kolom ke kolom berikutnya untuk menyelesaikan loop, maka dari itu untuk mengurutkan 7 nilai dibutuhkan 7 x loop atau sebanyak 49 x proses berpindah, bisa juga disebut (n x n) (n= jumlah data/nilai), Maka dari itu bisa disimpulkan bahwa big O nya adalah O(n^2).

- Jika n=1 maka: O(1^2)= 1
- Jika n=2 maka: O(2^2)= 4
- Jika n=3 maka: O(3^2)= 9
- Jika n=4 maka: O(4^2)= 16
- Jika n=5 maka: O(5^2)= 25

Kompleksitas selection sort berdasarkan jumlah data



Hasil run:

```
Masukkan jumlah data: 5
Data ke-1 = 4

Data ke-2 = 2

Data ke-3 = 3

Data ke-4 = 5

Data ke-5 = 1

Deret array yang sudah disortir: [1][2][3][4][5]
--------------------------------
Process exited after 15.09 seconds with return value 0
Press any key to continue . . .
```