

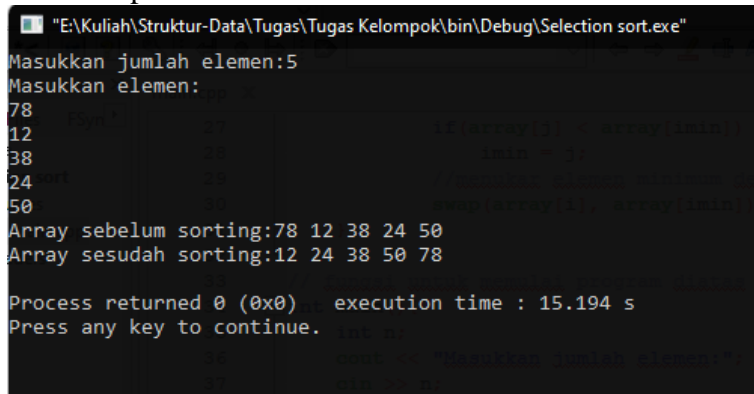
Nama : Rafie Mirza Ramadhan
NIM : 21091397037
Kelas : 2021A Manajemen Informatika

LAPORAN SELECTION SORT

1. Selection Sort

```
main.cpp X
1  #include<iostream>
2  using namespace std;
3
4  //fungsi untuk menukar isi dari variabel a dengan variabel b
5  void swapping(int &a, int &b){
6      int temp;
7      temp = a;
8      a = b;
9      b = temp;
10     // int a adalah untuk angka pertama
11     // int b adalah untuk angka kedua
12     // temp adalah tempat sementara bagi variabel yang sudah ditukar
13 }
14 // fungsi untuk menampilkan hasil array yang nantinya sudah selesai disortir oleh program
15 void display(int *array, int size){
16     for(int i = 0; i<size; i++){
17         cout << array[i] << " ";
18     }
19     cout << endl;
20 }
21 // fungsi untuk melakukan selection sort
22 void selectionSort(int *array, int size){
23     int i, j, imin;
24     for(i = 0; i<size-1; i++){
25         imin = i; //menjadi elemen minimum dari array yang tidak beraturan
26         for(j = i+1; j<size; j++){
27             if(array[j] < array[imin])
28                 imin = j;
29             //menukar elemen minimum dengan elemen pertama
30             swap(array[i], array[imin]);
31         }
32     }
33     // fungsi untuk memulai program diatas
34     int main(){
35         int n;
36         cout << "Masukkan jumlah elemen:"; // memasukkan jumlah elemen yang diinginkan
37         cin >> n;
38         int arr[n]; //membuat array dengan jumlah elemen yang diberikan
39         cout << "Masukkan elemen:" << endl;
40         for(int i = 0; i<n; i++){
41             cin >> arr[i]; // dimasukkannya elemen yang sudah diinputkan oleh user kedalam array
42         }
43         // untuk memunculkan array yang belum disorting
44         cout << "Array sebelum sorting:";
45         display(arr, n);
46         selectionSort(arr, n); // dilakukannya penyortiran array
47     }
48 }
49
50
main.cpp X
27     if(array[j] < array[imin])
28         imin = j;
29     //menukar elemen minimum dengan elemen pertama
30     swap(array[i], array[imin]);
31 }
32 }
33 // fungsi untuk memulai program diatas
34 int main(){
35     int n;
36     cout << "Masukkan jumlah elemen:"; // memasukkan jumlah elemen yang diinginkan
37     cin >> n;
38     int arr[n]; //membuat array dengan jumlah elemen yang diberikan
39     cout << "Masukkan elemen:" << endl;
40     for(int i = 0; i<n; i++){
41         cin >> arr[i]; // dimasukkannya elemen yang sudah diinputkan oleh user kedalam array
42     }
43     // untuk memunculkan array yang belum disorting
44     cout << "Array sebelum sorting:";
45     display(arr, n);
46     selectionSort(arr, n); // dilakukannya penyortiran array
47     cout << "Array sesudah sorting:";
48     display(arr, n); // fungsi untuk memunculkan array yang sudah disortir
49 }
50 }
```

2. Hasil Output



```
"E:\Kuliah\Struktur-Data\Tugas\Tugas Kelompok\bin\Debug\Selection sort.exe"
Masukkan jumlah elemen:5
Masukkan elemen:
78
12
38
24
50
Array sebelum sorting:78 12 38 24 50
Array sesudah sorting:12 24 38 50 78

Process returned 0 (0x0)   execution time : 15.194 s
Press any key to continue.
```

3. Kelebihan dan kekurangan selection sort:

a. Kelebihan:

- Algoritma ini sangat rapat dan mudah untuk diimplementasikan.
- Mempercepat pencarian.
- Mudah menentukan data maksimum / minimum.

b. Kekurangan:

- Membutuhkan metode tambahan.
- Sulit untuk digabungkan kembali.
- Perlu dihindari untuk penggunaan data lebih dari 1000 tabel, karena akan menyebabkan kompleksitas yang lebih tinggi dan kurang praktis.