**LAPORAN INDIVIDU  
SELECTION SORT**



**ACHMAD NURS SYURURI ARIFIN  
21091397024  
D4 MANAJEMEN INFORMATIKA**

1. **Kodingan Selection Sort**

#include<iostream>

using namespace std;

void swapping(int &a, int &b){ //menukar isi dari a dan b

int temp;

temp = a;

a = b;

b = temp;

}

void display(int \*array, int size){

for(int i = 0; i<size; i++)

cout << array[i] << " ";

cout << endl;

}

void selectionSort(int \*array, int size){

int i, j, imin;

for(i = 0; i<size-1; i++){

imin = i; //mendapatkan indeks data minimum

for(j = i+1; j<size; j++)

if(array[j] < array[imin])

imin = j;

//menempatkan di posisi yang benar

swap(array[i], array[imin]);

}

}

int main(){

int n;

cout << "Masukkan jumlah elemen:";

cin >> n;

int arr[n]; //membuat array dengan jumlah elemen yang diberikan

cout << "Masukkan elemen:" << endl;

for(int i = 0; i<n; i++){

cin >> arr[i];

}

cout << "Array sebelum sorting:";

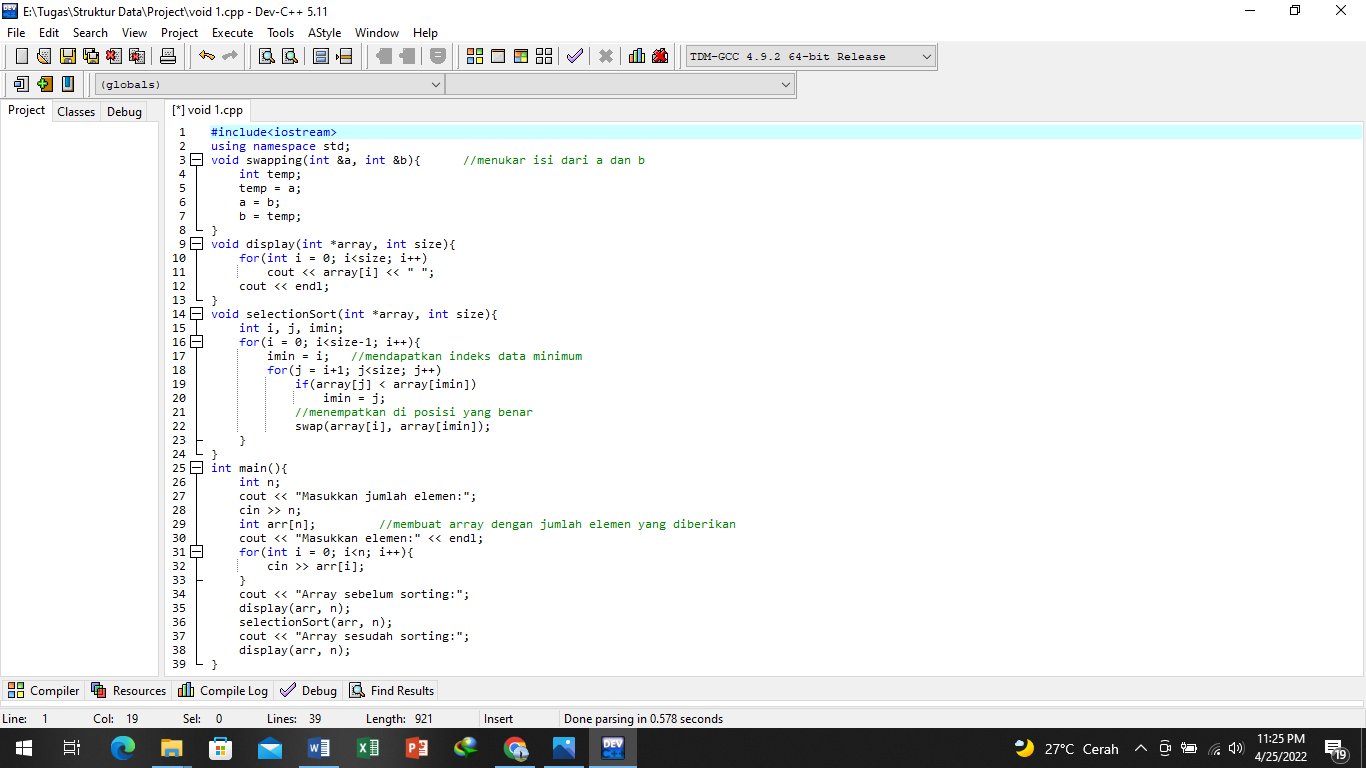
display(arr, n);

selectionSort(arr, n);

cout << "Array sesudah sorting:";

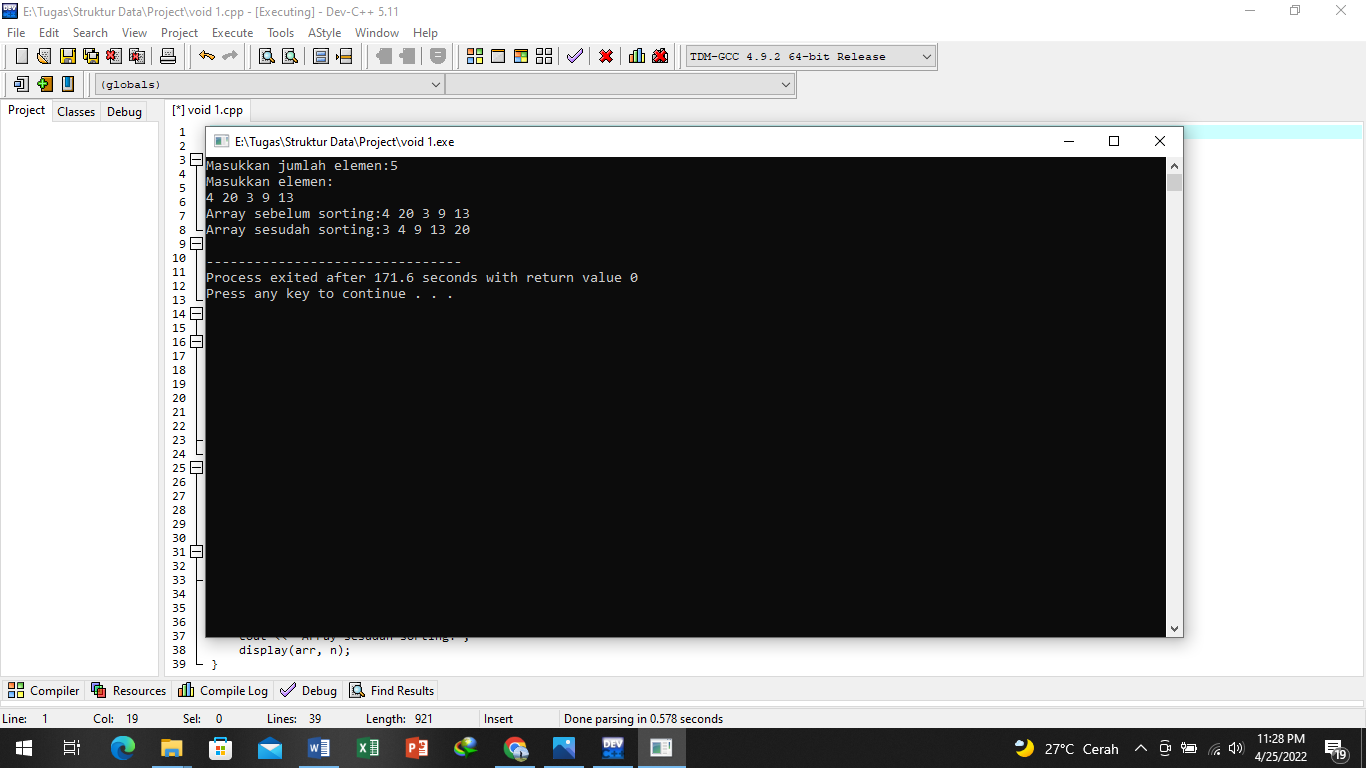
display(arr, n);

}



Dalam teknik Selection Sort, daftar dibagi menjadi dua bagian. Di satu bagian semua elemen diurutkan dan di bagian lain item tidak disortir. Pada awalnya kita mengambil data maksimum atau minimum dari array. Setelah mendapatkan data (misalnya minimum) kita menempatkannya di awal daftar dengan mengganti data tempat pertama dengan data minimum. Setelah melakukan array semakin kecil. Demikian teknik ini dilakukan.

1. **Hasil Run Kodingan Selection Sort**



1. **Menghitung Jenis Big O**

void display(int \*array, int size){

for(int i = 0; i<size; i++)

cout << array[i] << " ";

cout << endl;

}

void selectionSort(int \*array, int size){

int i, j, imin;

for(i = 0; i<size-1; i++){

imin = i; //mendapatkan indeks data minimum

for(j = i+1; j<size; j++)

if(array[j] < array[imin])

imin = j;

//menempatkan di posisi yang benar

swap(array[i], array[imin]);

Berdasarkan kodingan diatas, berapa kali cout array akan dijalankan dengan nilai n sebagai berikut

* n = 1, maka akan dijalankan 1 kali
* n = 5, maka akan dijalankan 5 kali
* n = 10, maka akan dijalankan 10 kali
* dan seterusnya…

sehingga cout array akan dijalankan sebanyak

Dalam menghitung time complexity dan space complexity (yang biasanya dinotasikan dengan Big-O notation), ada beberapa aturan yang perlu dilakukan disini:

* abaikan konstanta, misalkan O(N + 2), maka dianggap O(n) saja.
* abaikan non dominant terms,misalkan O(N² + N), maka dianggap O(n²) saja.

maka, time complexity nya adalah .

1. **Kelebihan dan Kekurangan Selection Sort**

Setiap metode pengerjaan pastilah memiliki kekurangan dan kelebihannya dalam prosesnya, seperti halnya lebih cepat namun tidak praktis ataupun lebih lama namun prosesnya sangat praktis.

Begitupun dengan algoritma Selection Sort yang memiliki kekurangan dan kelebihan dalam proses pengerjannya.

Kelebihan Selection Sort :

a) Algoritma ini sangat rapat dan mudah untuk diimplementasikan.

b) Mempercepat pencarian

c) Mudah menentukan data maksimum /minimum.

d) Mudah menggabungkannya kembali.

e) Kompleksitas selection sort relatif lebih kecil.

Kekurangan Selection Sort :

a) Membutuhkan method tambahan

b) Sulit untuk digabungkan kembali

c) Perlu dihindari untuk penggunaan data lebih dari 1000 tabel, karena akan menyebabkan kompleksitas yang lebih tinggi dan kurang praktis