

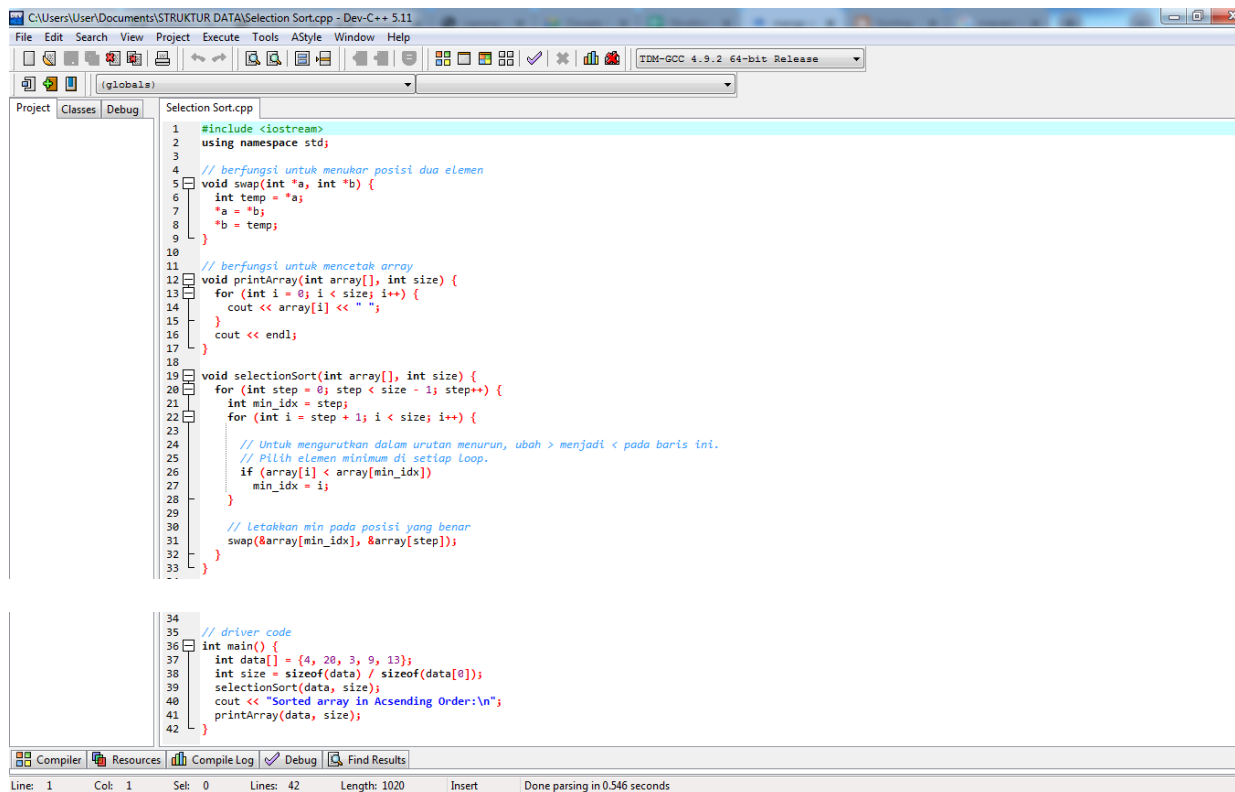
Nama : Shinta Berlina Maharani

NIM : 21091397048

Manajemen Informatika 2021B

SELECTION SORT

1. Laporan kodingan, screenshot kode tipe sorting yang anda buat, beri penjelasan, dan bukti berupa screenshot hasil run kodingan

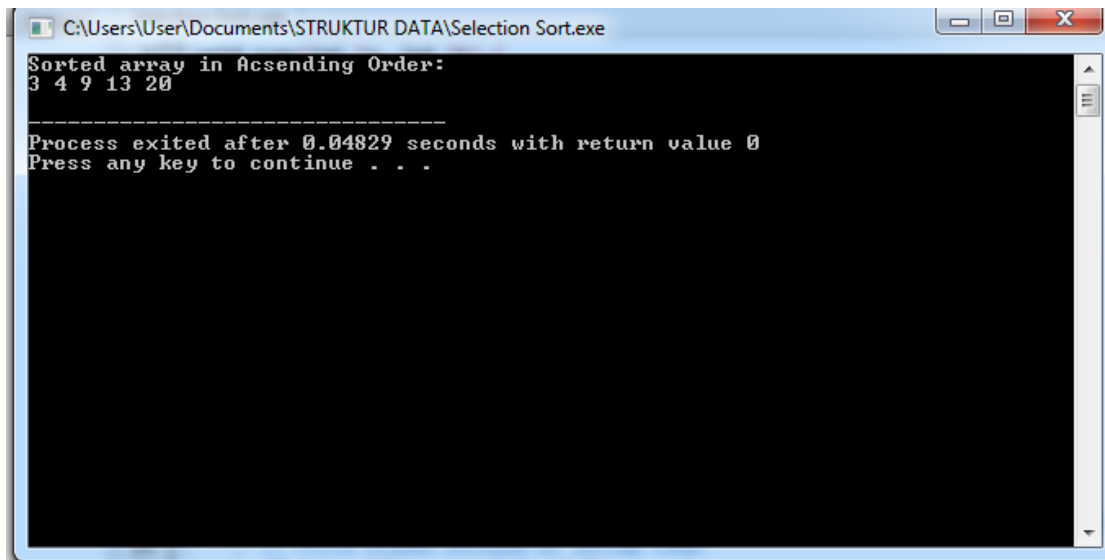


The screenshot shows a C++ code editor window titled "C:\Users\User\Documents\STRUKTUR DATA\Selection Sort.cpp - Dev-C++ 5.11". The code implements the Selection Sort algorithm. It includes necessary headers, defines a swap function, a printArray function, and the selectionSort function. The main function initializes an array with values {4, 20, 3, 9, 13} and calls the selectionSort function. The output of the program is "Sorted array in Ascending Order:\n".

```
1 #include <iostream>
2 using namespace std;
3
4 // berfungsi untuk menukar posisi dua elemen
5 void swap(int *a, int *b) {
6     int temp = *a;
7     *a = *b;
8     *b = temp;
9 }
10
11 // berfungsi untuk mencetak array
12 void printArray(int array[], int size) {
13     for (int i = 0; i < size; i++) {
14         cout << array[i] << " ";
15     }
16     cout << endl;
17 }
18
19 void selectionSort(int array[], int size) {
20     for (int step = 0; step < size - 1; step++) {
21         int min_idx = step;
22         for (int i = step + 1; i < size; i++) {
23             // Untuk mengurutkan dalam urutan menurun, ubah > menjadi < pada baris ini.
24             // Pilih elemen minimum di setiap loop.
25             if (array[i] < array[min_idx])
26                 min_idx = i;
27         }
28         // letakkan min pada posisi yang benar
29         swap(&array[min_idx], &array[step]);
30     }
31 }
32
33
34 // driver code
35 int main() {
36     int data[] = {4, 20, 3, 9, 13};
37     int size = sizeof(data) / sizeof(data[0]);
38     selectionSort(data, size);
39     cout << "Sorted array in Ascending Order:\n";
40     printArray(data, size);
41 }
42
```

Line: 1 Col: 1 Sel: 0 Lines: 42 Length: 1020 Insert Done parsing in 0.546 seconds

Hasil Run :



```
C:\Users\User\Documents\STRUKTUR DATA\Selection Sort.exe
Sorted array in Ascending Order:
3 4 9 13 20
-----
Process exited after 0.04829 seconds with return value 0
Press any key to continue . . .
```

Selection sort adalah suatu metode pengurutan yang membandingkan elemen yang sekarang dengan elemen berikutnya sampai ke elemen yang terakhir. Jika ditemukan elemen lain yang lebih kecil dari elemen sekarang maka dicatat posisinya dan langsung ditukar. Metode selection sort adalah melakukan pemilihan dari suatu nilai yang terkecil dan kemudian menukarnya dengan elemen paling awal, lalu membandingkan dengan elemen yang sekarang dengan elemen berikutnya sampai dengan elemen terakhir, perbandingan dilakukan terus sampai tidak ada lagi pertukaran data.

Konsep Selection Sort

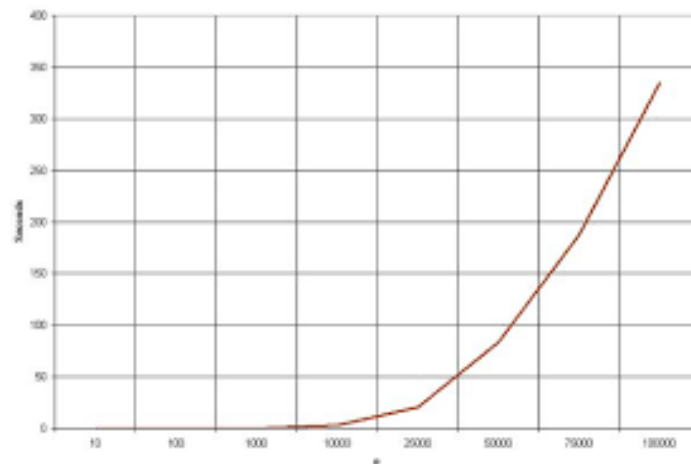
Algoritma pengurutan sederhana salah satunya adalah Selection Sort. Ide dasarnya adalah melakukan beberapa kali pass untuk melakukan penyeleksian elemen struktur data. Untuk sorting ascending (menaik), elemen yang paling kecil di antara elemen-elemen yang belumurut, disimpan indeksinya, kemudian dilakukan pertukaran nilai elemen dengan indeks yang disimpan tersebut dengan elemen yang paling depan yang belumurut. Sebaliknya, untuk sorting descending (menurun), elemen yang paling besar yang disimpan indeksinya kemudian ditukar. Selection Sort diakui karena kesederhanaan algoritmanya dan performanya lebih bagus daripada algoritma lain yang lebih rumit dalam situasi tertentu. Algoritma ini bekerja sebagai berikut: 1. Mencari nilai minimum (jika ascending) atau maksimum (jika descending) dalam sebuah list 2. Menukarkan nilai ini dengan elemen pertama list 3. Mengulangi langkah di atas untuk sisa list dengan dimulai pada posisi kedua. Secara efisien kita membagi list menjadi dua bagian yaitu bagian yang sudah diurutkan, yang didapat dengan membangun dari kiri ke kanan dan dilakukan pada saat awal, dan bagian list yang elemennya akan diurutkan.

Kompleksitas 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. Berdasarkan operasi perbandingan elemennya:

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

Grafik kompleksitas Selection Sort :



Analisa Selection Sort

Sama seperti algoritma pengurutan gelembung, algoritma ini mempunyai dua buah kalang, satu kalang di dalam kalang yang lainnya. Banyaknya perbandingan yang harus dilakukan untuk siklus pertama adalah n, perbandingan yang harus dilakukan untuk siklus yang kedua n-1, dan seterusnya. Sehingga jumlah keseluruhan perbandingan adalah $n(n+1)/2-1$ perbandingan. Menggunakan algoritma pengurutan seleksi, hindari pengurutan nilai dengan data pada tabel lebih besar dari 1000 buah, dan hindari mengurutkan tabel lebih dari beberapa ratus kali. Berikut Dengan interval data antara 100 sampai dengan 1.000 elemen. Waktu eksekusi diukur dengan satuan Second (s).

Waktu eksekusi algoritma selection sort

Jumlah elemen <i>array</i>	Waktu (s)
100	0.87
250	5.59
500	21.54
750	48.62
1000	66.86

Kelebihan Selection Sort :

1. Algoritma ini sangat rapat dan mudah untuk diimplementasikan.
2. Mempercepat pencarian.
3. Mudah menentukan data maksimum.
4. Mudah menentukan data minimum.
5. Kompleksitas selection sort relative lebih kecil.

Kekurangan Selection Sort :

1. Membutuhkan method tambahan.
2. Sulit untuk digabungkan kembali.
3. Perlu dihindari untuk data lebih dari 1000 tabel, karena akan menyebabkan kompleksitas yang lebih tinggi dan kurang praktis.