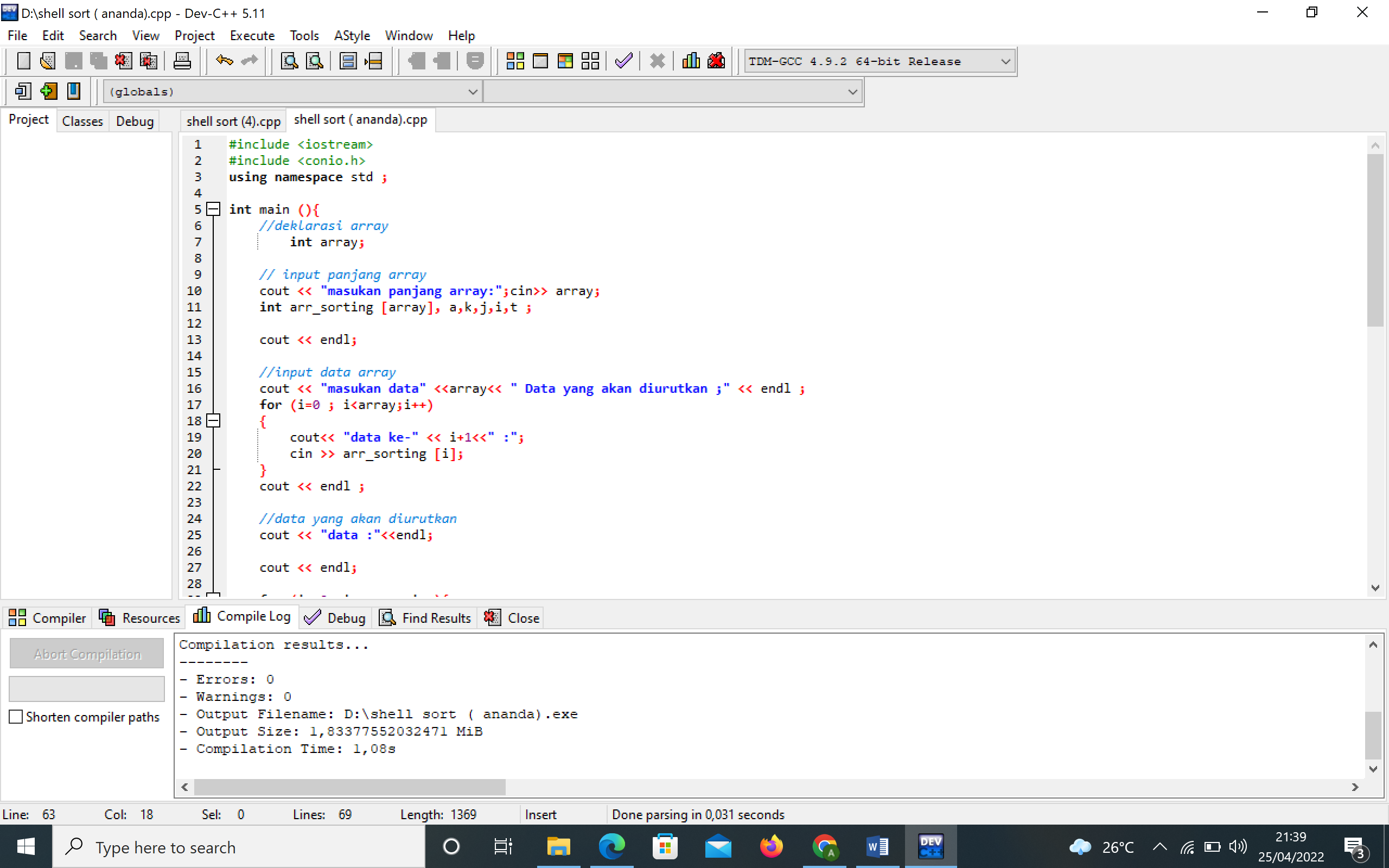
**NAMA : ANANDA PUTRI RAHMADANI**

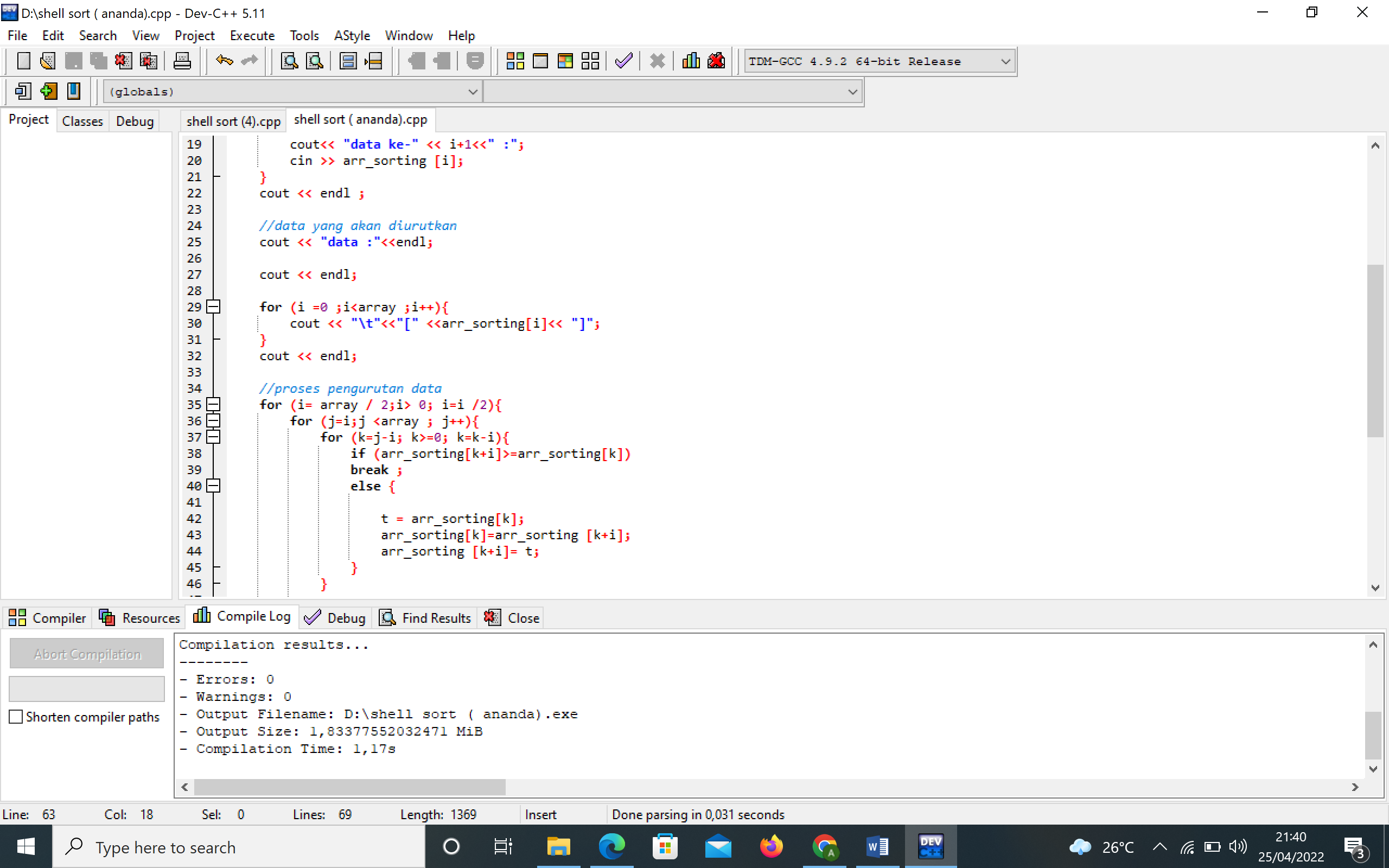
**NIM : 21091397046**

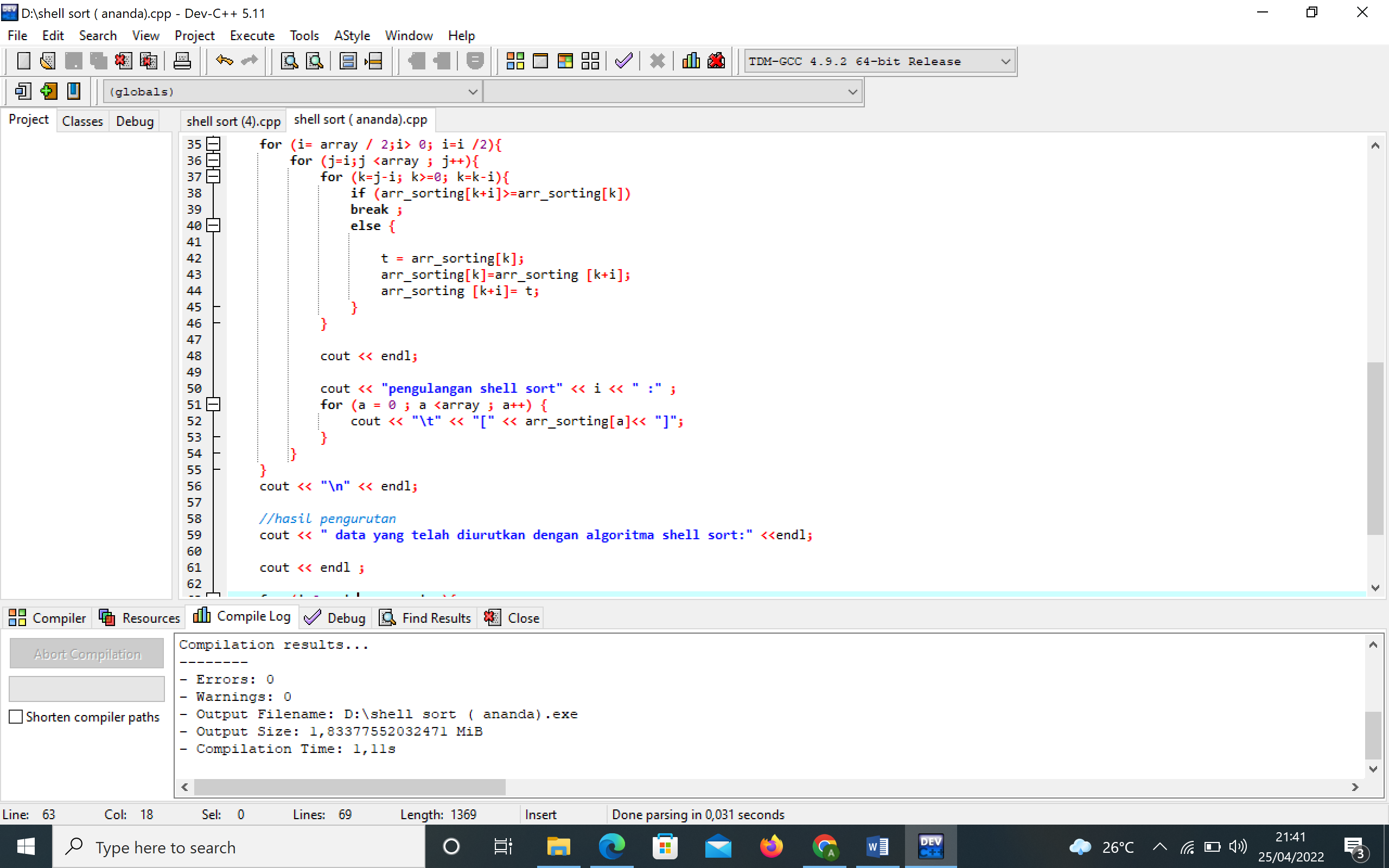
**D4 MANAJEMEN INFORMATIKA**

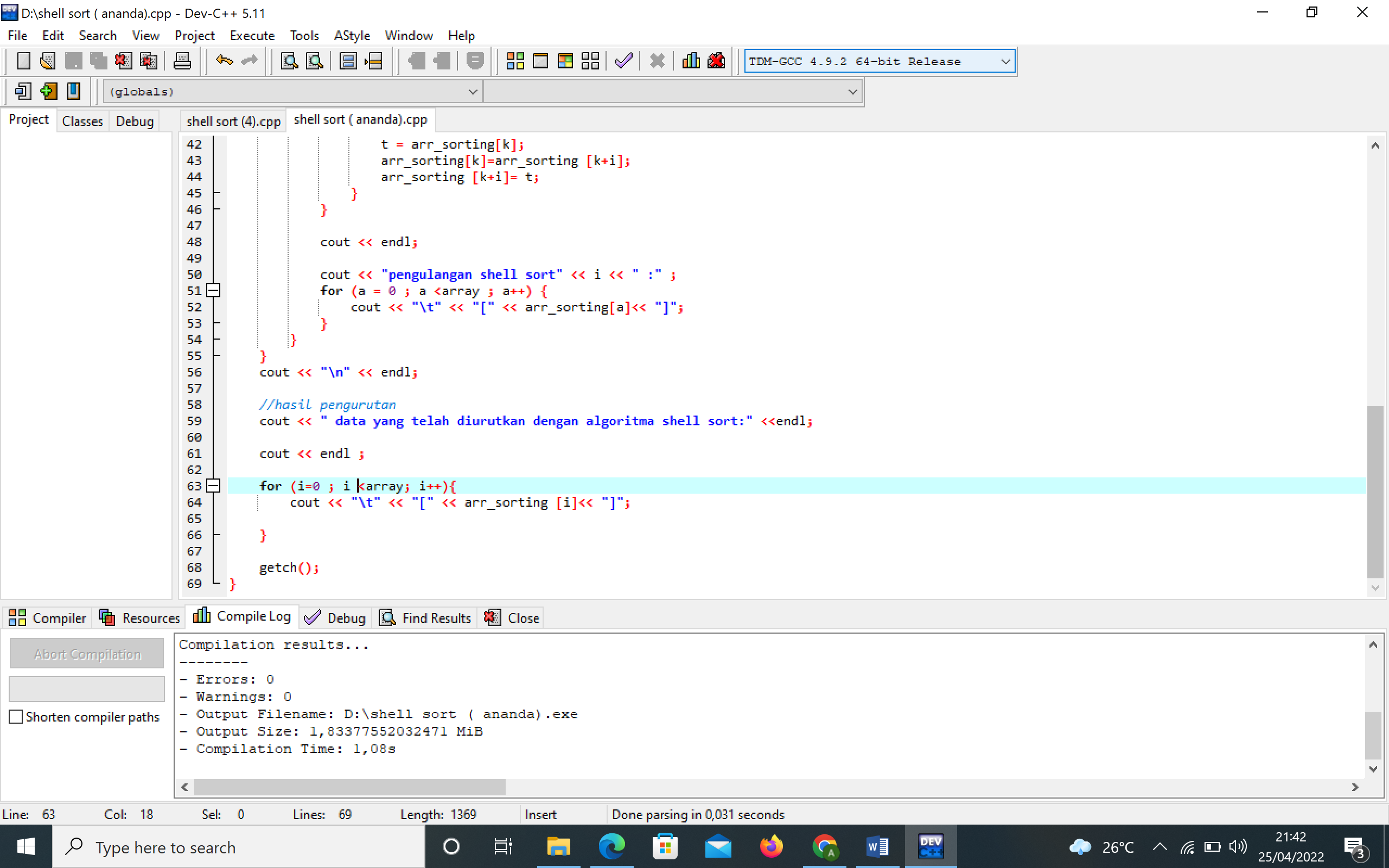
**STRUKTUR DATA ( Laporan Individu SHELL SORT)**

* Laporan kodingan , screenshot kode tipe sorting yang anda buat , beri penjelasan , dan bukti berupa screenshot hasil run kodingan
* Screenshot kode tipe sorting yang dibuat









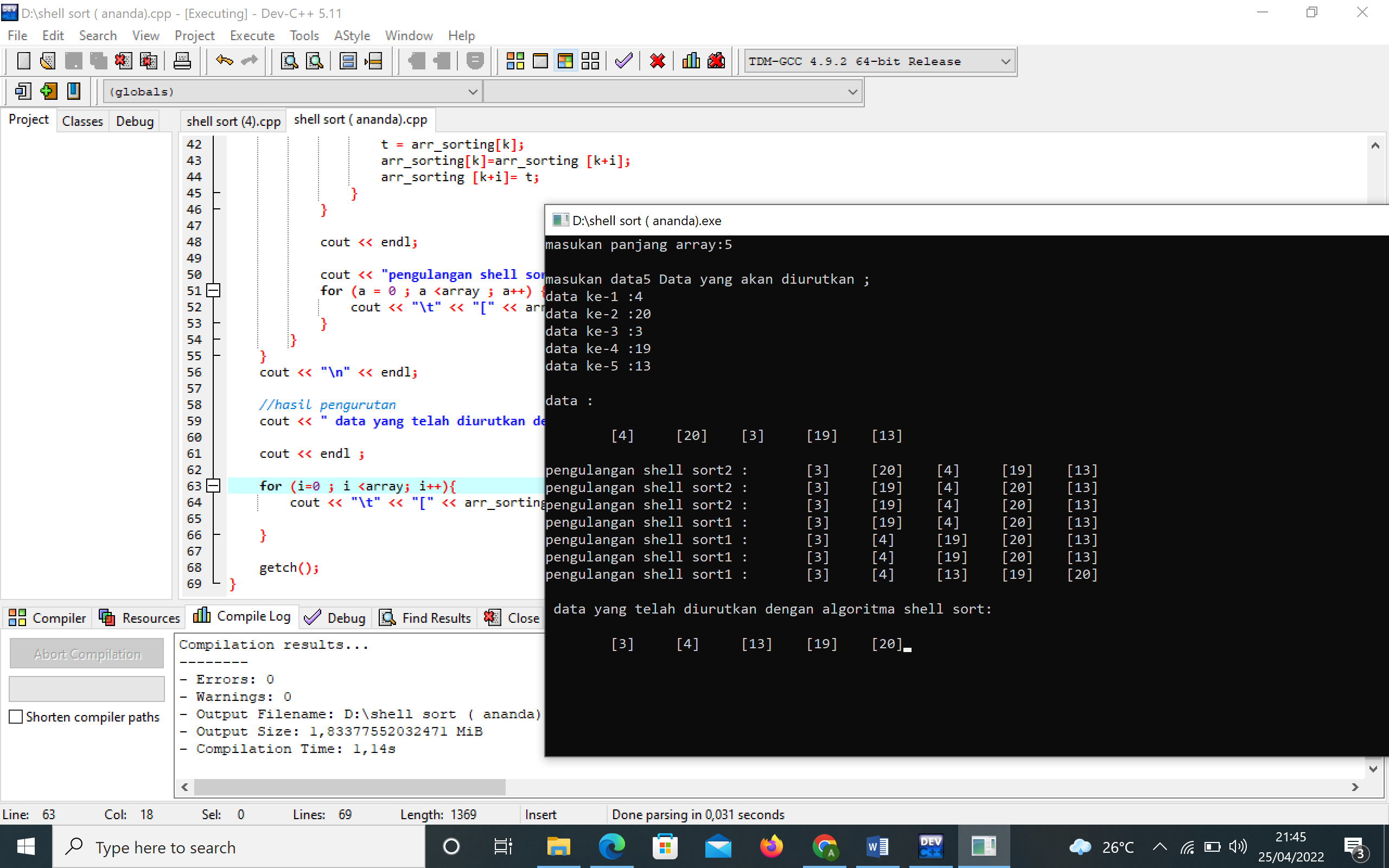
* Penjelasan Algoritma Shell Sort

1. Progam akan dijalankan jika range indeks > 0 terpenuhi
2. Menentukan jarak data yang akan dibandingkan dengan banyak data (n) dibagi 2 , jika hasilnya bilangan ganjil maka hilangkan 0,5nya ( dibulatkan) . Misalkan banyak data adalah 7 maka jaraknya 7/2 = 3,5 dibulatkan menjadi 3
3. Misal data dikiri adalah x dan data di kanan adalah y maka , jika x > y = salah , maka data tersebut tidak dapat ditukar dan jika x >y = benar maka data dapat ditukar
4. Jika jarak < 1 , maka pengurutan akan berhenti

* Penjelasan kodingan

|  |  |
| --- | --- |
| //deklarasi array | Komentar deklarasi array |
| Int array | Mendeklarasikan variable array dengan tipe integer |
| // input panjang array  cout << "masukan panjang array:";cin>> array;  int arr\_sorting [array], a,k,j,i,t ; | -Menginput Panjang array  -Mendeklarasikan variable a,k,j,I,t , arr\_sorting dengan tipe integer  -Tanda kurung siku dipakai untuk menandai array |
| cout << "masukan data" <<array<< " Data yang akan diurutkan ;" << endl ;  for (i=0 ; i<array;i++)  {  cout<< "data ke-" << i+1<<" :";  cin >> arr\_sorting [i];  } | -Memasukan nilai dari arr\_sorting atau memasukan data yang ingin diurutkan  - i=0 pengulangan dimulai dari indeks 0 |
| cout << "data :"<<endl;    cout << endl;    for (i =0 ;i<array ;i++){  cout << "\t"<<"[" <<arr\_sorting[i]<< "]";  } | -Menampilkan data yang akan diurutkan  -Menampilkan nilai variable arr\_sorting pada indeks dari i |
| for (i= array / 2;i> 0; i=i /2){ | Pengulangan array |
| for (i= array / 2;i> 0; i=i /2){  for (j=i;j <array ; j++){  for (k=j-i; k>=0; k=k-i){  if (arr\_sorting[k+i]>=arr\_sorting[k])  break ;  else {    t = arr\_sorting[k];  arr\_sorting[k]=arr\_sorting [k+i];  arr\_sorting [k+i]= t;  }  }    cout << endl;    cout << "pengulangan shell sort" << i << " :" ;  for (a = 0 ; a <array ; a++) {  cout << "\t" << "[" << arr\_sorting[a]<< "]";  }  }  }  cout << "\n" << endl; | Proses pengurutan dengan shell sort |
| j = i | Nilai awal j adalah i |
| J<array | Pengulangan dilakukan selama j kurang dari array |
| k= j-i | Nilai awal k adalah j dikurangi i |
| if (arr\_sorting[k+i]>=arr\_sorting[k])  break ; | Percabangan if jika syarat terpenuhi maka perulangan berhenti |
| cout << "pengulangan shell sort" << i << " :" ; | Pengulangan kesekian dengan pembaginya |
| for (a = 0 ; a <array ; a++) {  cout << "\t" << "[" << arr\_sorting[a]<< "]";  } | Menampilkan proses pengurutan |
| for (i=0 ; i <array; i++){  cout << "\t" << "[" << arr\_sorting [i]<< "]";  } | Menampilkan hasil pengurutan |
| getch () | Memiliki fungsi untuk menahan tampilan |

* Bukti berupa screenshot hasil run kodingan (output)



* Hitung jenis big O (worst case scenario untuk time complexity)nya , jelaskan kompleksitasnya adalah yang anda temukan
* Hitung Big O saat n = 1
* Hitung Big O saat n = 5
* Hitung Big O saat n=10
* Jelaskan menurut anda Big Onya termasuk jenis yang mana dan jelaskan kenapa
* Jelaskan kelebihan dan kekurangan sorting yang kalian buat dibandingkan yang dibuat oleh teman kalian

- Kelebihan :

1. Algoritma shell sort sangat rapat dan mudah untuk diimplementasikan

2. Operasi pertukaranya hanya dilakukan sekali saja

3. Waktu pengurutan bisa lebih ditekan

4. Mudah menggabungkannya kembali

5. Kompleksitas shell sort relative lebih kecil

- Kekurangan :

1. Memperlukan method tambahan

2. Sulit untuk membagi masalah