**LAPORAN INDIVIDU STRUKTUR DATA**

**“INSERTION SORT”**



ALVIN NOOR HIDAYAH

21091397016

D4 MANAJEMEN INFORMATIKA

1. **Codingan insertion sort**

#include <iostream>

using namespace std;

int main(){

int y;

cout<<"masukan banyak array:";

cin>>y;

int x[y];

// disini itu inputan dari user dimana variabel y banyak data//

// dan array X wadah untuk data yang akan diinputkan//

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

cout<<"masukan angka ke "<<i<<" :";

//disini saya melakukan perulangan untuk mengisi index

// dari setiap array mulai dari array ke 0//

cin>>x[i];

cout<<endl;

}

for(int i=1; i<y; i++){

int key = x[i];

int j = i-1;

while(j>=0 && x[j] > key){

x[j+1] = x[j];

j--;

//disini pertama akan mulai perulangan

//dari indeks ke 1 yaitu variabel i

//variabel i akan menajdi kunci untuk dibandingkan sebelumnya

//jika bilangan sebelumnya lebih besar, maka key akan kembali

//key kembali dibandingkan dengan bil. ke 2 hingga menemukan

// sampai key menemukan bilangan yang diinginkan

//untuk menghasilkan nilai shortingan yang diinginkan disini nilai harus berada disebelah paling kanan//

//yaitu harus di variabel key tersebut//

}

x[j+1] = key;

cout<<"proses sorting"<<endl;

//disni akan terjadi proses sort yang akan menghitung angka//

//kemudian melakukan proses sorting angka yang diinginkan//

for(int m=0;m<y;m++){

cout<<x[m]<<" ";

}

cout<<endl;

}

cout<<"hasil akhir"<<endl;

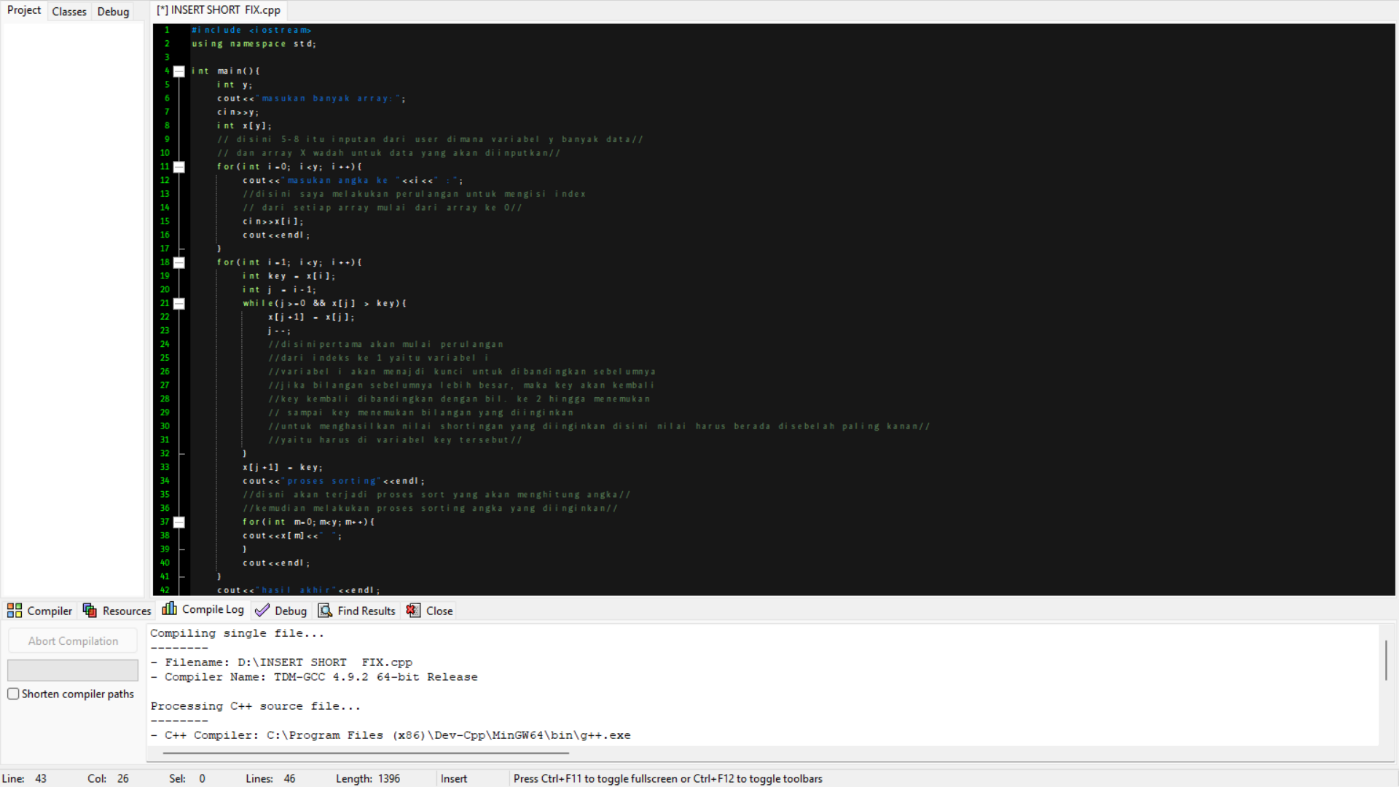
for(int m=0;m<y;m++){

cout<<”[“<<x[m]<<" ]";

}

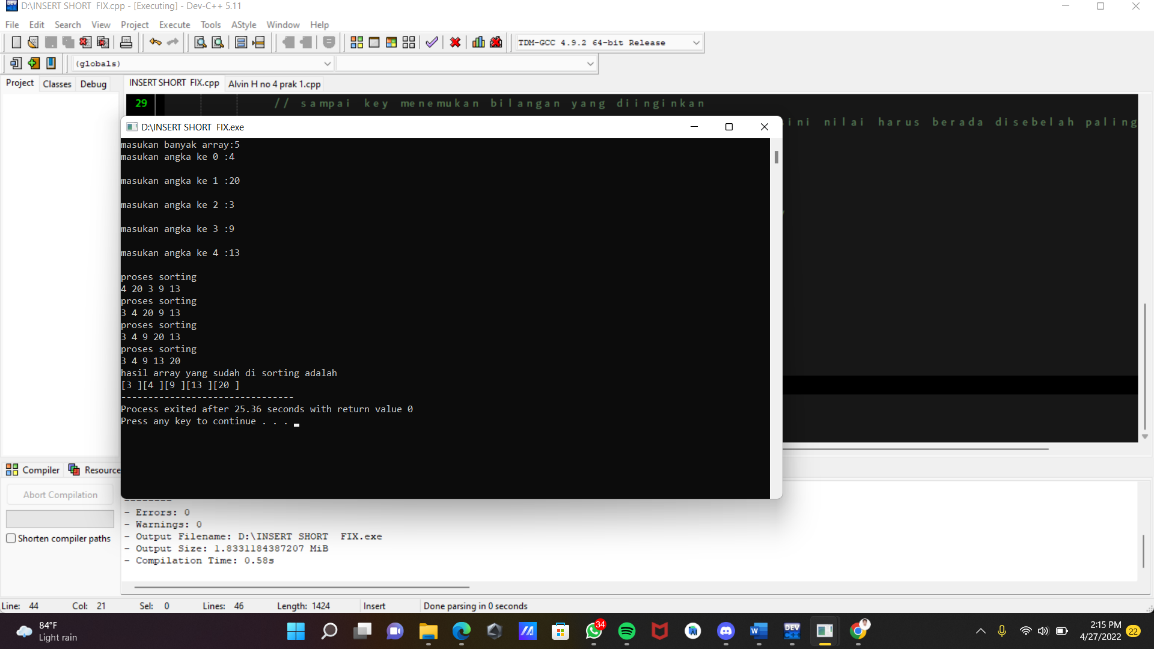
}

1. **Contoh codingan**



Insertion sort ialah teknik mengurut data dengan membandingkan dan mengisih dua item data pertama pada tata susunan, kemudian membandingkan data tatasusunan bersebelahan dengan perkara yang sedia ada. Isihan sisipan algoritma ialah proses mengurut kartu di tangan kita. Algoritma ini mampu menyusun data dari besar ke kecil (menaik) dan dari kecil ke besar (menurun). Algoritma ini mungkin tidak sesuai untuk set data dengan banyak item kerana kerumitan algoritma boleh menjadi sangat menggalakkan. Insertion Sort algoritma yang lebih kompleks, karena membutuhkan wadah Data sementara, dalam rangka untuk beradaptasi dengan data, data sebelumnya Kemudian ada proses geser(tidak bergerak)untuk mencapai urutan pengolahan, Oleh karena itu, proses waktu yang relatif lebih lama dalam proses acak(langsung bergerak.)

1. **Hasil Run codingan insertion sort**

****

1. **Kelebihan dan kelemahan insertion sort**

Kelebihan:

* Salah satu cara tercepat
* algoritma pengurutan cocok untuk menyortir dengan banyak data
* karena hal itu daftar on the spot. Ini tidak memerlukan memori tambahan untuk menjalankannya.

Kekurangan;

* Algoritma ini kurang efisien jika data telah diurutkan sebelumnya. Ada lebih banyak angka bulat yang lebih cocok untuk diurutkan.
* Menemukan posisi yang tepat untuk elemen dalam array dapat memerlukan sejumlah operasi. Ini tidak praktis untuk array yang begitu besar.
* Jika daftar diurutkan dengan cara yang salah, setiap eksekusi perintah harus memindai dan mengganti seluruh bagian sebelum memasukkan elemen berikutnya.