**LAPORAN PRATIKUM STRUKTUR DATA**

**JOBSHEET 8**



**Disusun Oleh :**

**NAJWA ELDIARA OWILIA TIKSA**

**BP/NIM : 2023/23343079**

**Dosen Pengampu :**

**Randi Proska Sandra, S.Pd, M.Sc**

**Kode Kelas : 202323430157**

**PROGRAM STUDI S1 INFORMATIKA**

**DEPARTEMEN ELEKTRONIKA**

**FAKULITAS TEKNIK**

**UNIVERSITAS NEGERI PADANG**

**2023**

**Bubble Sort**

1. **Source Code**

//Created by 23343079\_Najwa Eldiara Owilia Tiksa

#include <stdio.h>

void bubbleSort(int arr[], int n) {

int i, j, temp;

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

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

if (arr[j] > arr[j+1]) {

// Tukar posisi elemen jika urutan salah

temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

}

int main() {

int arr[] = {64, 34, 25, 12, 22, 11, 90};

int n = sizeof(arr)/sizeof(arr[0]);

int i;

printf("Array sebelum diurutkan: \n");

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

printf("%d ", arr[i]);

printf("\n");

bubbleSort(arr, n);

printf("Array setelah diurutkan: \n");

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

printf("%d ", arr[i]);

printf("\n");

return 0;

}

1. **Penjelasan Program**

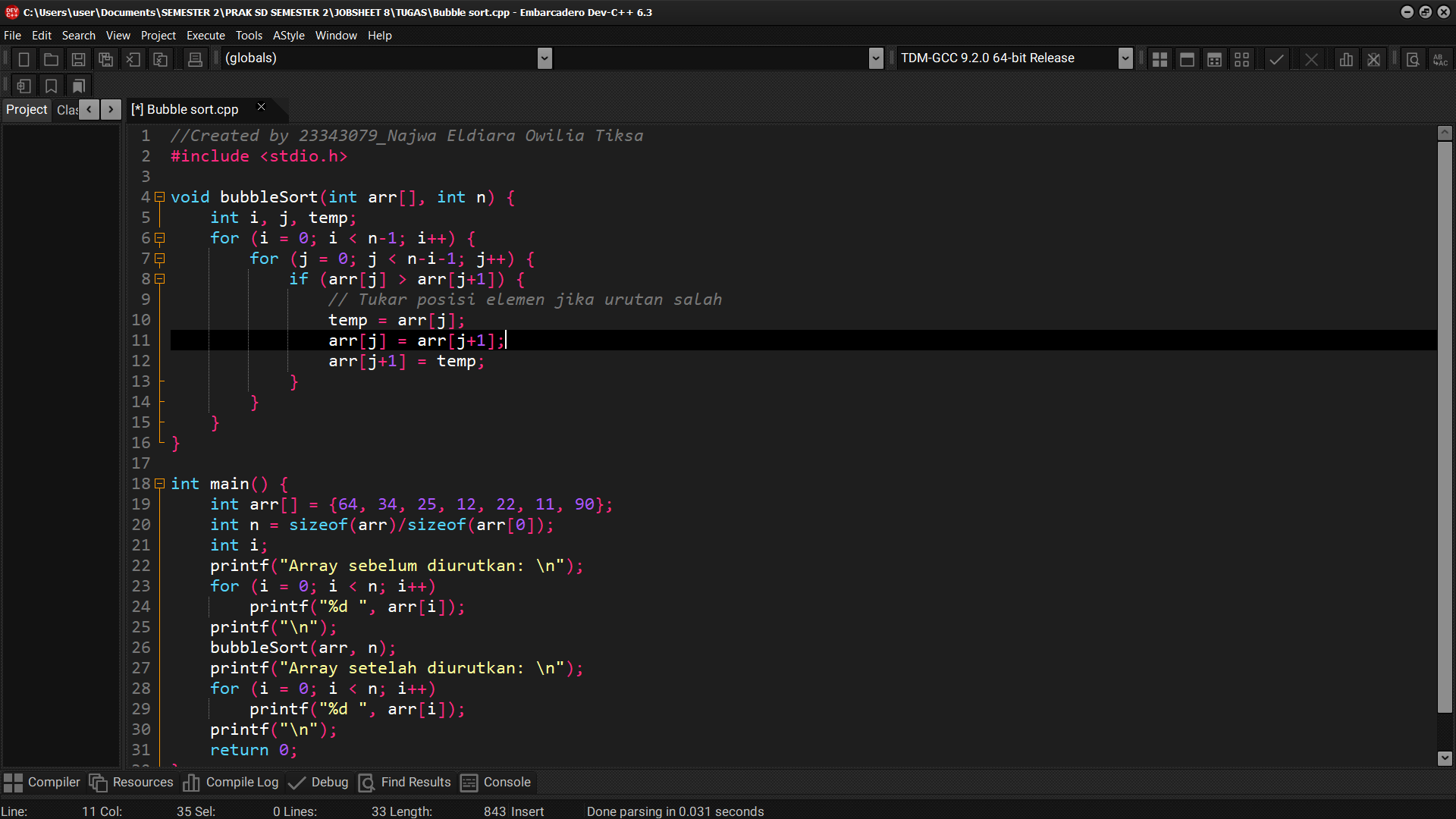
Program di atas adalah implementasi algoritma Bubble Sort dalam bahasa C. Algoritma Bubble Sort adalah salah satu algoritma pengurutan sederhana yang bekerja dengan cara membandingkan setiap elemen secara berpasangan dan menukar posisi jika diperlukan sehingga elemen-elemen yang lebih besar akan "menggelembung" ke atas. Proses ini diulang sampai tidak ada lagi pertukaran yang perlu dilakukan, sehingga array menjadi terurut.

Program memiliki fungsi `bubbleSort` yang menerima array dan ukuran array sebagai parameter. Fungsi ini mengurutkan array yang diberikan menggunakan algoritma Bubble Sort. Di dalam fungsi `bubbleSort`, terdapat dua loop bersarang. Loop pertama digunakan untuk iterasi melalui setiap elemen array kecuali elemen terakhir. Loop kedua digunakan untuk membandingkan setiap elemen dengan elemen di sebelah kanannya. Jika elemen saat ini lebih besar dari elemen di sebelah kanannya, maka kedua elemen tersebut ditukar posisinya.Proses pertukaran ini diulang sampai tidak ada lagi pertukaran yang diperlukan.

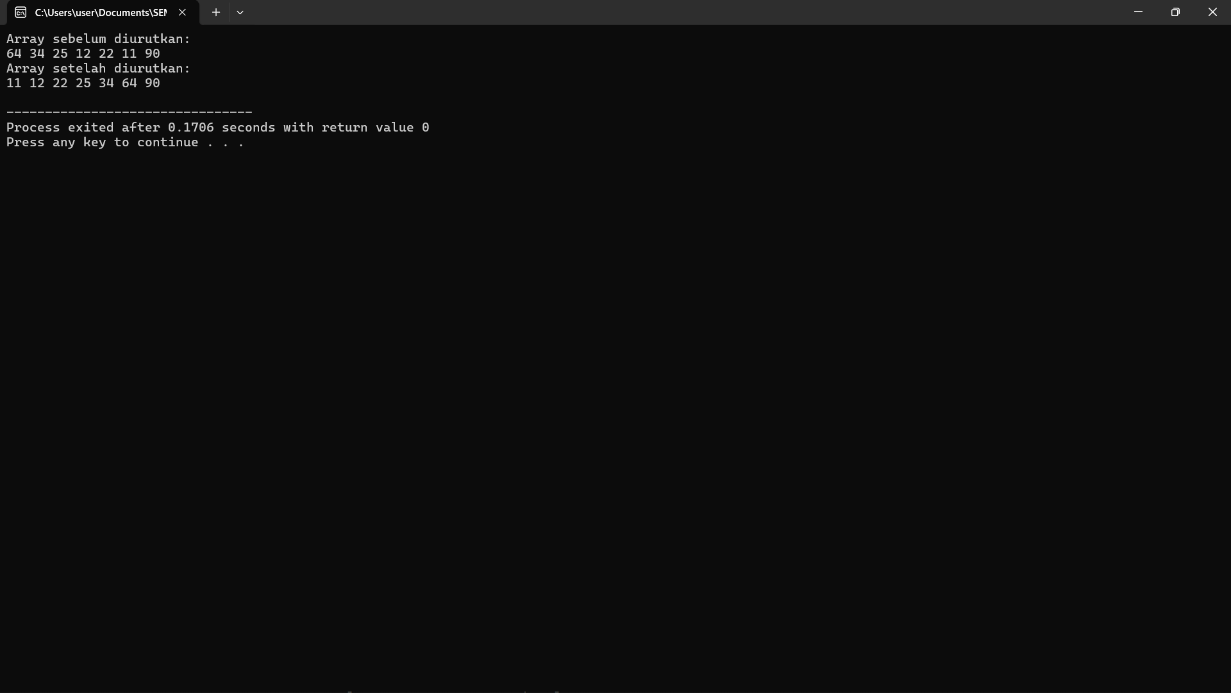
Di dalam `main` function, sebuah array integer didefinisikan dengan beberapa nilai awal.

Array tersebut kemudian diurutkan menggunakan fungsi `bubbleSort`. Hasil pengurutan kemudian dicetak ke layar.

1. **Screenshot program**



1. **Output**

****

**Insertion Sort**

1. **Source Code**

// Created by 23343079\_Najwa Eldiara Owilia Tiksa

#include <stdio.h>

#include <string.h>

#define MAX\_KONTAK 100

// Struktur data untuk menyimpan kontak

struct Kontak {

char nama[50];

char nomor[15];

};

// Fungsi untuk menampilkan daftar kontak

void tampilkanKontak(struct Kontak kontak[], int n) {

printf("Daftar Kontak:\n");

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

printf("Nama: %s\t Nomor: %s\n", kontak[i].nama, kontak[i].nomor);

}

}

// Fungsi insertion sort untuk mengurutkan daftar kontak berdasarkan nama

void insertionSort(struct Kontak kontak[], int n) {

int i, j;

struct Kontak key;

for (i = 1; i < n; i++) {

key = kontak[i];

j = i - 1;

// Pindahkan elemen dari kontak[0..i-1] yang lebih besar dari key ke posisi satu lebih tinggi dari posisi saat ini

while (j >= 0 && strcmp(kontak[j].nama, key.nama) > 0) {

kontak[j + 1] = kontak[j];

j = j - 1;

}

kontak[j + 1] = key;

}

}

int main() {

struct Kontak daftarKontak[MAX\_KONTAK];

int jumlahKontak;

printf("Masukkan jumlah kontak: ");

scanf("%d", &jumlahKontak);

// Membaca data kontak dari pengguna

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

printf("Masukkan nama kontak ke-%d: ", i + 1);

scanf(" %[^\n]s", daftarKontak[i].nama);

printf("Masukkan nomor kontak ke-%d: ", i + 1);

scanf("%s", daftarKontak[i].nomor);

}

// Menampilkan daftar kontak sebelum diurutkan

printf("Daftar Kontak sebelum diurutkan:\n");

tampilkanKontak(daftarKontak, jumlahKontak);

// Mengurutkan daftar kontak menggunakan insertion sort

insertionSort(daftarKontak, jumlahKontak);

// Menampilkan daftar kontak setelah diurutkan

printf("\nDaftar Kontak setelah diurutkan:\n");

tampilkanKontak(daftarKontak, jumlahKontak);

return 0;

}

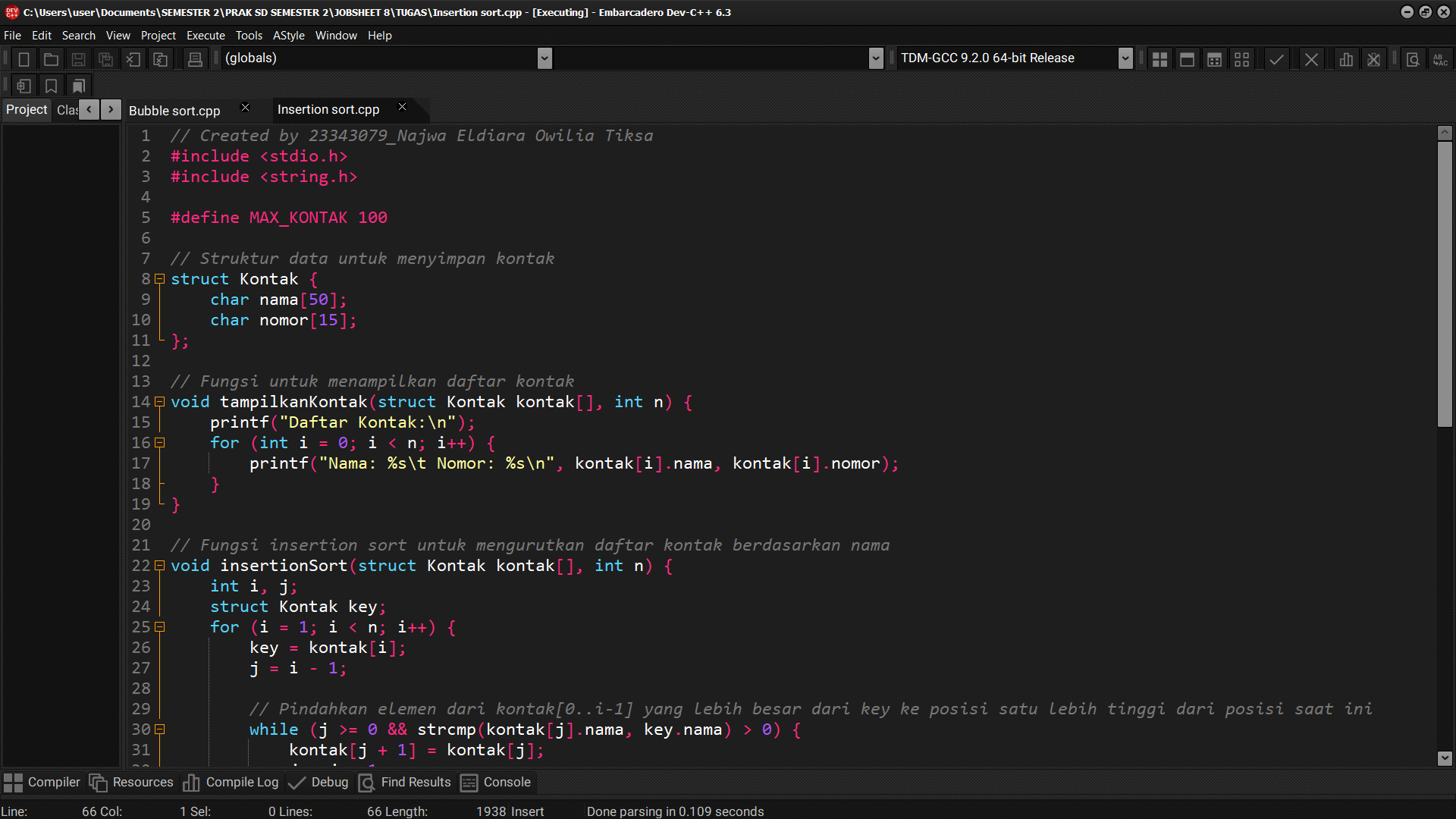
1. **Penjelasan Program**

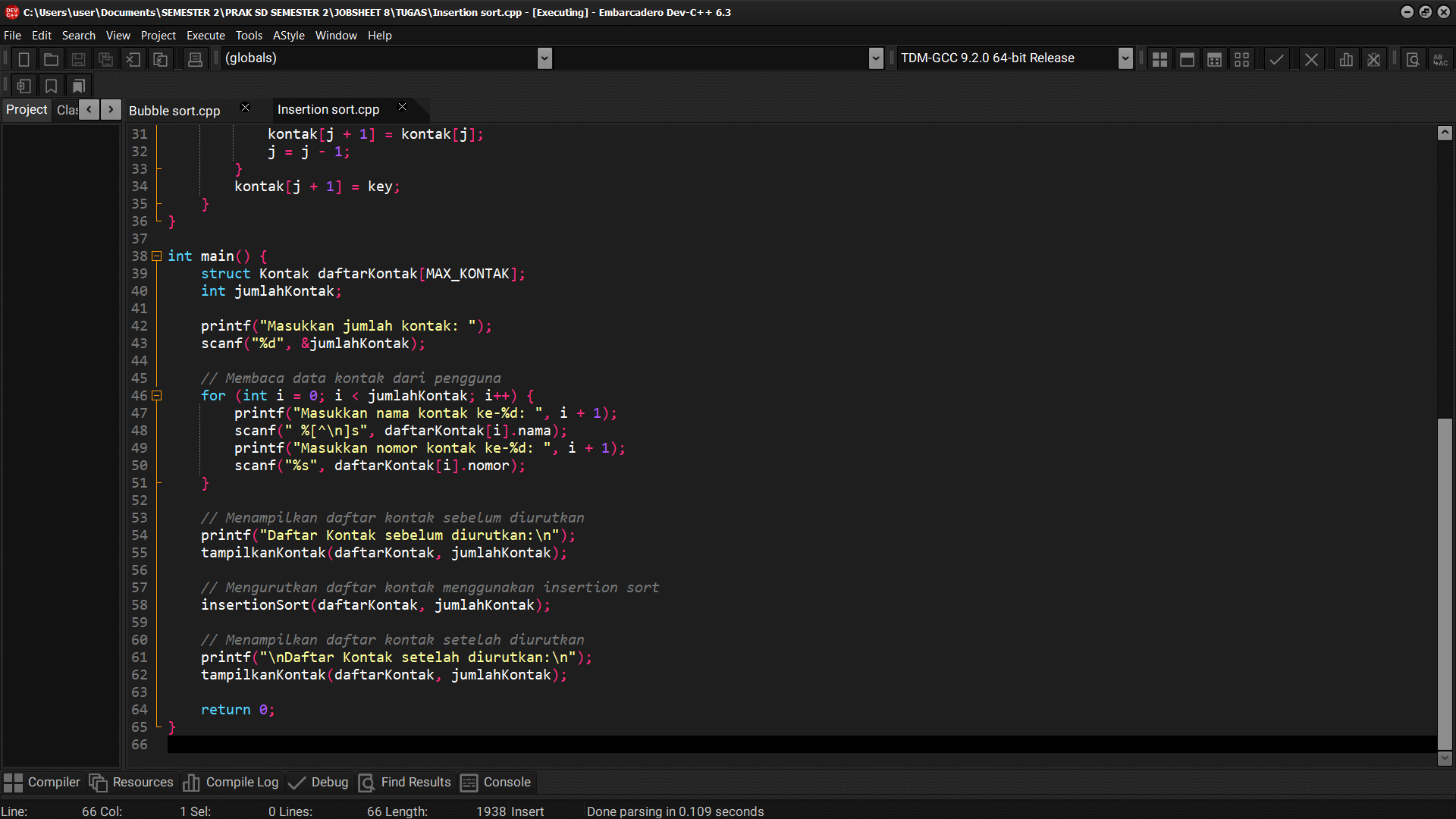
Program ini adalah sebuah aplikasi sederhana untuk menyimpan daftar kontak (nama dan nomor telepon) dan mengurutkannya berdasarkan nama menggunakan algoritma insertion sort.

Fungsi Insertion Sort (`insertionSort`) digunakan untuk mengurutkan array kontak berdasarkan nama. Algoritma ini bekerja dengan membandingkan setiap elemen dengan elemen sebelumnya dan menukar posisi mereka jika diperlukan sehingga elemen-elemen tersebut terurut secara berurutan.

Algoritma yang digunakan dalam program ini adalah insertion sort, yang cukup efisien untuk jumlah kontak yang relatif kecil (hingga beberapa ratus atau ribu). Ini memindahkan elemen satu per satu melalui array, membandingkan setiap elemen dengan elemen-elemen sebelumnya, dan menukar posisi mereka jika diperlukan, sehingga elemen-elemen tersebut terurut secara berurutan.

1. **Screenshot program**





1. **Output**

