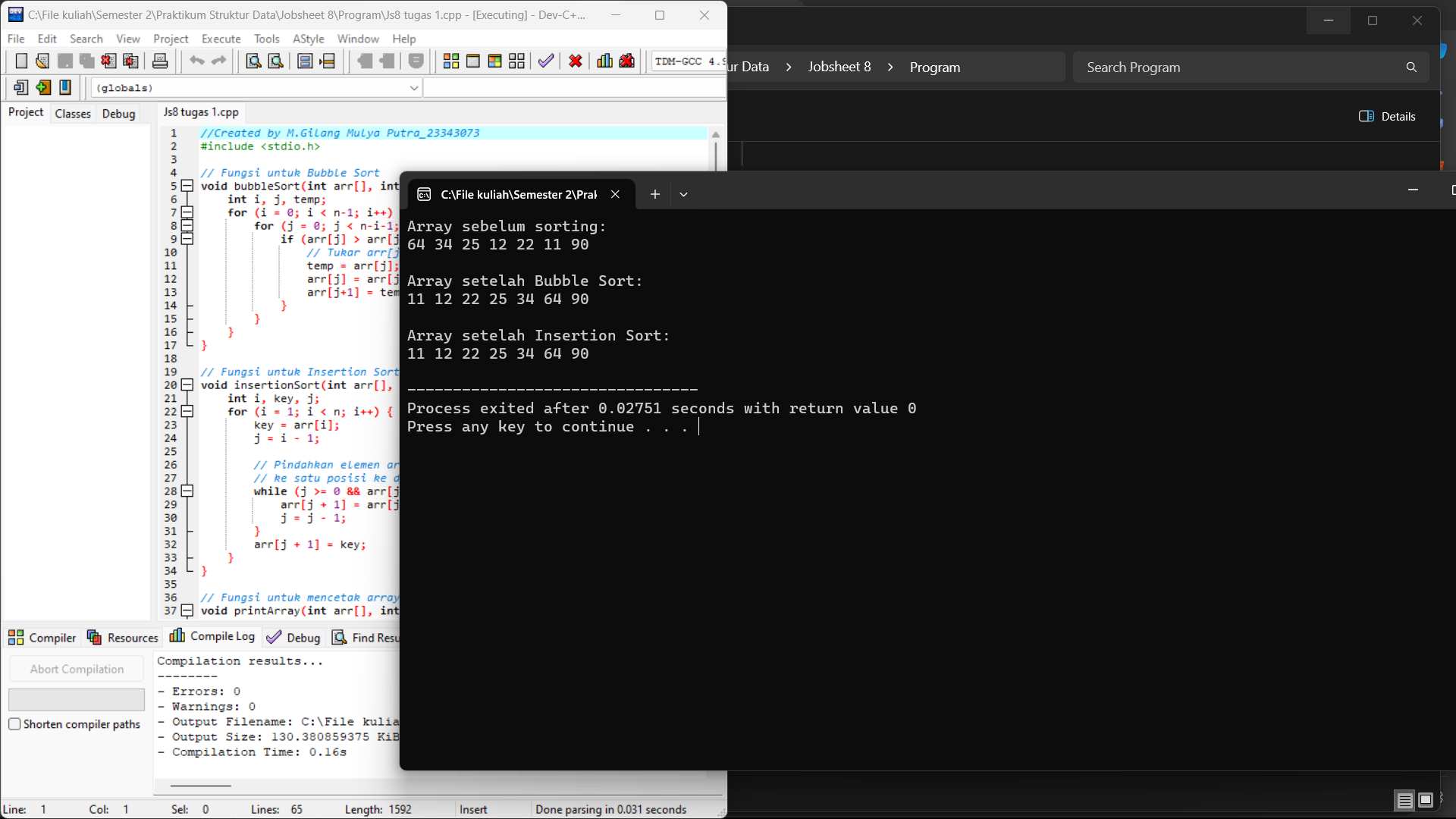
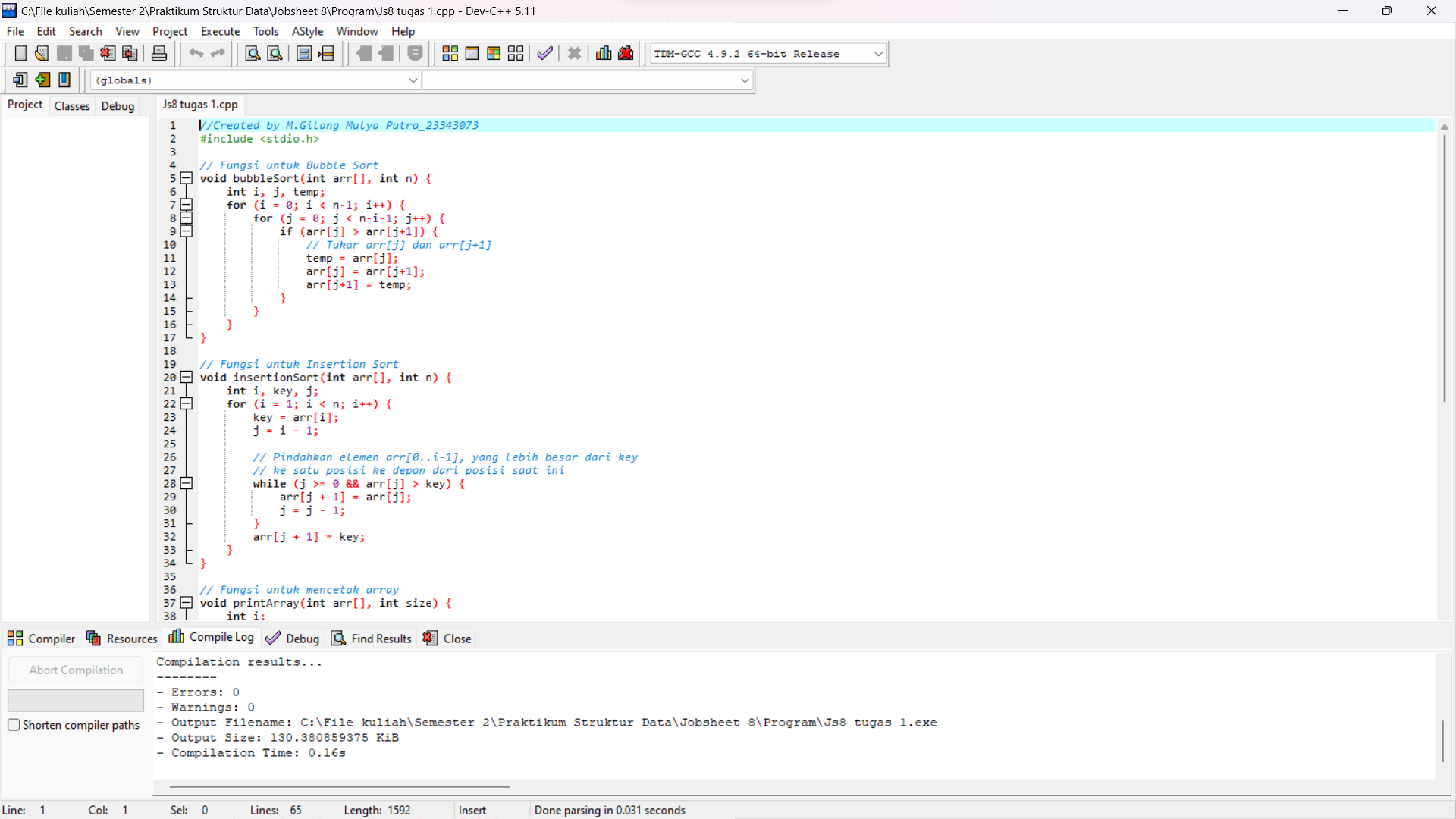
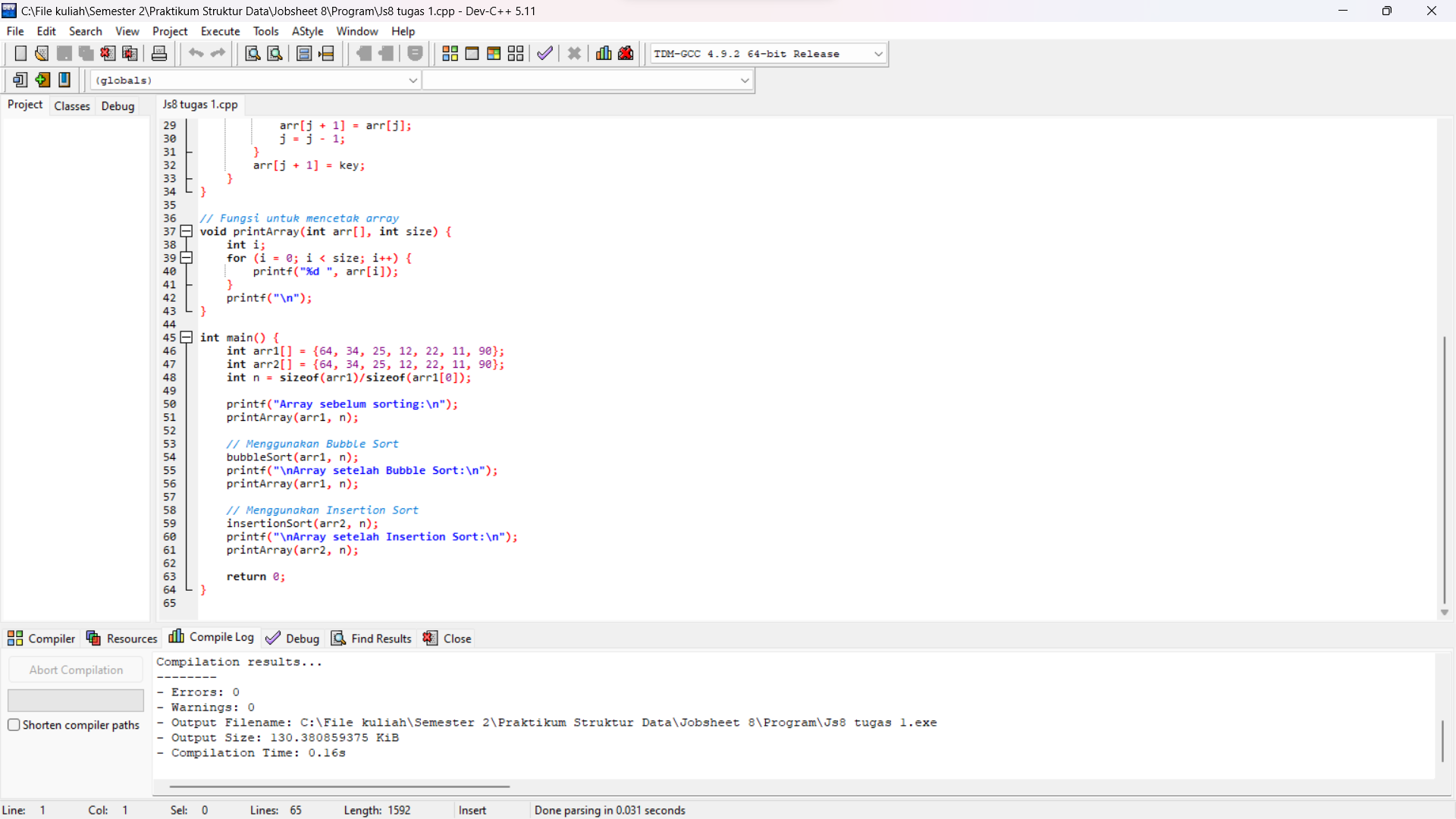
Nama : M.Gilang Mulya Putra

NIM : 23343073

Prodi : Informatika







Berikut adalah contoh implementasi Bubble Sort dan Insertion Sort dalam bahasa C, beserta penjelasan cara kerja aplikasi tersebut:

Aplikasi C

```c

#include <stdio.h>

// Fungsi untuk Bubble Sort

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 arr[j] dan arr[j+1]

temp = arr[j];

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

arr[j+1] = temp;

}

}

}

}

// Fungsi untuk Insertion Sort

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

int i, key, j;

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

key = arr[i];

j = i - 1;

// Pindahkan elemen arr[0..i-1], yang lebih besar dari key

// ke satu posisi ke depan dari posisi saat ini

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

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

j = j - 1;

}

arr[j + 1] = key;

}

}

// Fungsi untuk mencetak array

void printArray(int arr[], int size) {

int i;

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

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

}

printf("\n");

}

int main() {

int arr1[] = {64, 34, 25, 12, 22, 11, 90};

int arr2[] = {64, 34, 25, 12, 22, 11, 90};

int n = sizeof(arr1)/sizeof(arr1[0]);

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

printArray(arr1, n);

// Menggunakan Bubble Sort

bubbleSort(arr1, n);

printf("\nArray setelah Bubble Sort:\n");

printArray(arr1, n);

// Menggunakan Insertion Sort

insertionSort(arr2, n);

printf("\nArray setelah Insertion Sort:\n");

printArray(arr2, n);

return 0;

}

```

Penjelasan

Bubble Sort

1. Prinsip Kerja:

- Bubble Sort bekerja dengan cara membandingkan elemen yang berdekatan dan menukarnya jika mereka berada dalam urutan yang salah.

- Setiap iterasi memastikan bahwa elemen terbesar yang belum terurut "menggelembung" ke posisi akhirnya.

2. Langkah-langkah:

- Loop pertama berjalan dari 0 hingga n-1.

- Loop kedua berjalan dari 0 hingga n-i-1 (mengurangi jumlah iterasi karena elemen terbesar sudah berada di posisi akhirnya).

- Jika elemen saat ini lebih besar dari elemen berikutnya, mereka ditukar.

3. elebihan dan Kekurangan:

- Sederhana untuk diimplementasikan.

- Tidak efisien untuk daftar besar (kompleksitas waktu O(n^2)).

Insertion Sort

1. Prinsip Kerja:

- Insertion Sort bekerja dengan cara mengambil satu elemen dari daftar yang belum terurut dan menempatkannya pada posisi yang benar dalam daftar yang sudah terurut.

2. Langkah-langkah:

- Loop berjalan dari elemen kedua hingga akhir array.

- Elemen saat ini disimpan sebagai `key`, dan elemen sebelumnya dibandingkan dengan `key`.

- Jika elemen sebelumnya lebih besar dari `key`, mereka digeser ke kanan untuk memberikan ruang bagi `key`.

- `Key` ditempatkan pada posisi yang benar setelah semua elemen yang lebih besar digeser.

3. Kelebihan dan Kekurangan

- Efisien untuk daftar kecil dan hampir terurut.

- Kompleksitas waktu terbaik adalah O(n), tetapi terburuk tetap O(n^2).

Kesimpulan

Implementasi Bubble Sort dan Insertion Sort di atas menunjukkan bagaimana kedua algoritma bekerja dalam bahasa C. Bubble Sort melakukan banyak penukaran dan memiliki kompleksitas waktu O(n^2), membuatnya kurang efisien untuk daftar besar. Insertion Sort juga memiliki kompleksitas waktu O(n^2) dalam kasus terburuk, tetapi lebih efisien untuk daftar kecil dan hampir terurut.