**LAPORAN PRATIKUM STRUKTUR DATA**

**JOBSHEET 5**



**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**

**Selection sort**

1. **Source Code**

//Created by 23343079\_Najwa Eldiara Owilia Tiksa

#include <stdio.h>

#define MAX\_SIZE 100

// Fungsi untuk menukar nilai dua variabel

void swap(int \*a, int \*b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

// Fungsi untuk menampilkan array

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

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

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

}

printf("\n");

}

// Fungsi selection sort

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

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

int min\_index = i;

for (int j = i + 1; j < size; j++) {

if (arr[j] < arr[min\_index]) {

min\_index = j;

}

}

swap(&arr[i], &arr[min\_index]);

}

}

int main() {

int arr[MAX\_SIZE];

int size;

printf("Masukkan jumlah elemen array: ");

scanf("%d", &size);

if (size <= 0 || size > MAX\_SIZE) {

printf("Ukuran array tidak valid.\n");

return 1;

}

printf("Masukkan %d elemen array:\n", size);

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

scanf("%d", &arr[i]);

}

printf("Array sebelum diurutkan: ");

printArray(arr, size);

// Panggil fungsi selection sort

selectionSort(arr, size);

printf("Array setelah diurutkan: ");

printArray(arr, size);

return 0;

}

1. **Penjelasan Program**

Program tersebut merupakan implementasi algoritma Selection Sort dalam bahasa pemrograman C. selection sort memilih elemen terkecil dari array dan menukarnya dengan elemen pertama. Kemudian, ia memilih elemen terkecil dari sisa array dan menukarnya dengan elemen kedua, dan begitu seterusnya.

- Pertama, program meminta pengguna untuk memasukkan jumlah elemen array.

- Kemudian, program meminta pengguna untuk memasukkan elemen-elemen array tersebut.

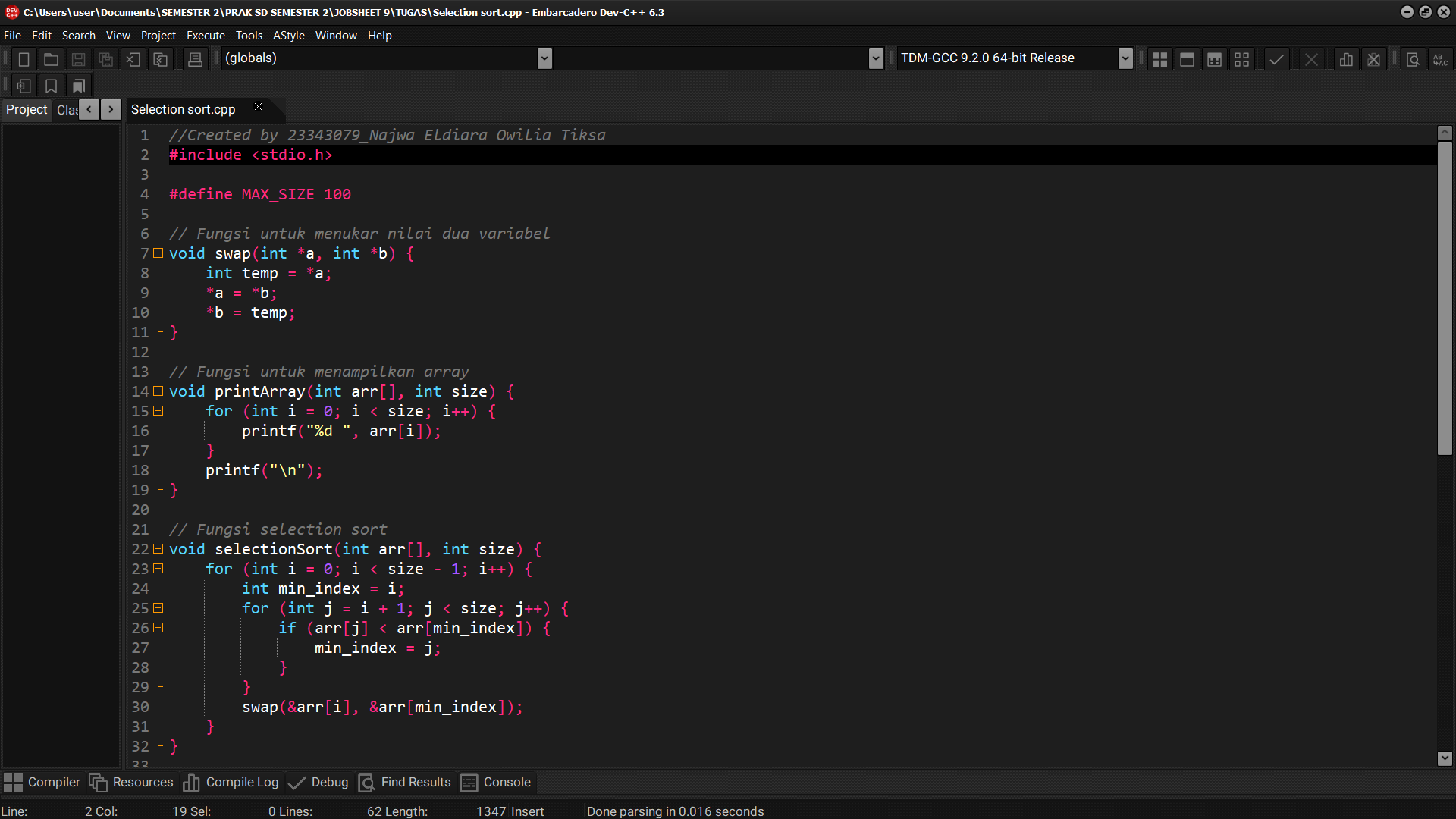
- Setelah itu, program mencetak array sebelum diurutkan.

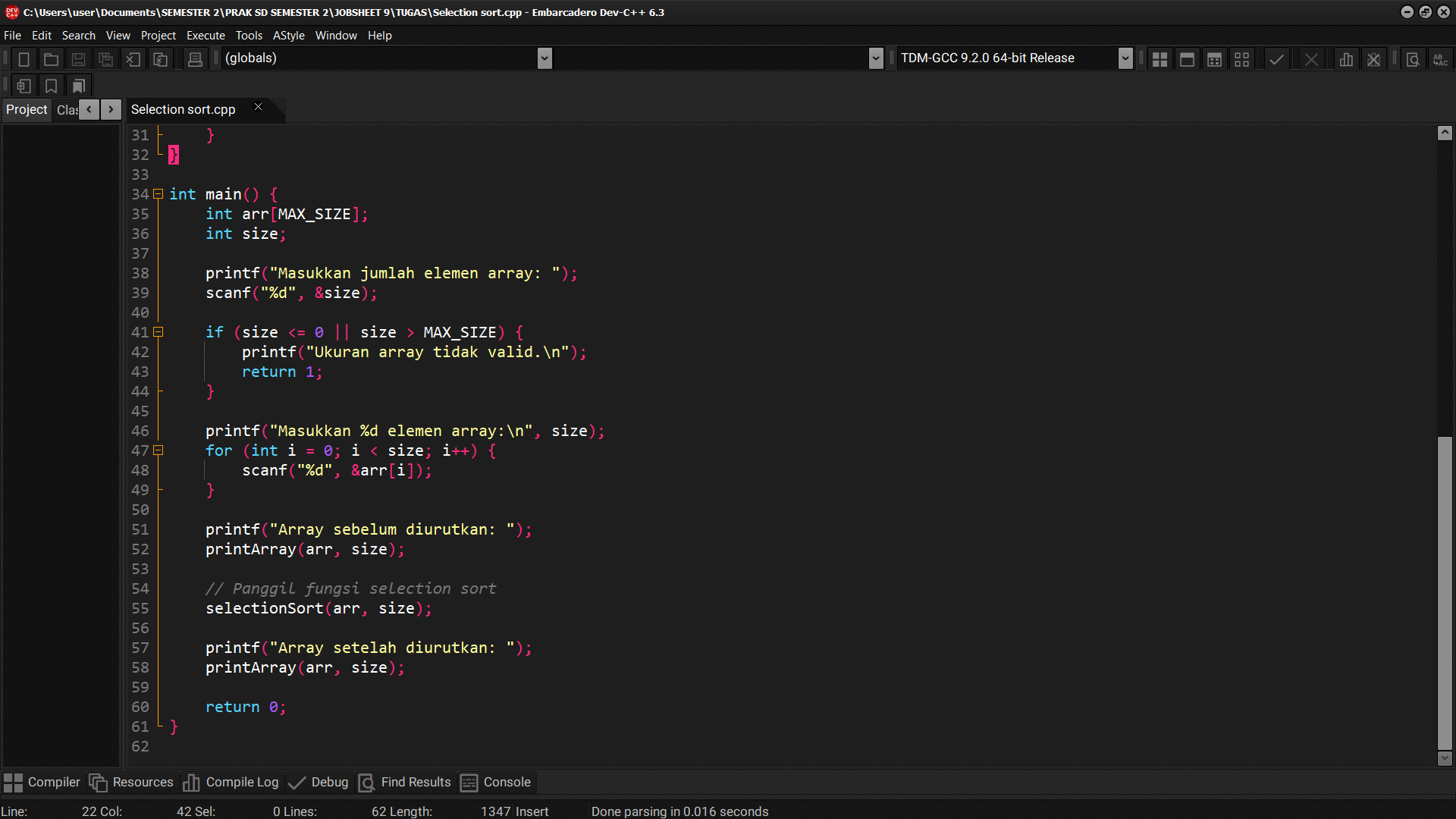
- Panggilan ke fungsi `selectionSort()` untuk mengurutkan array.

- Terakhir, program mencetak array setelah diurutkan.

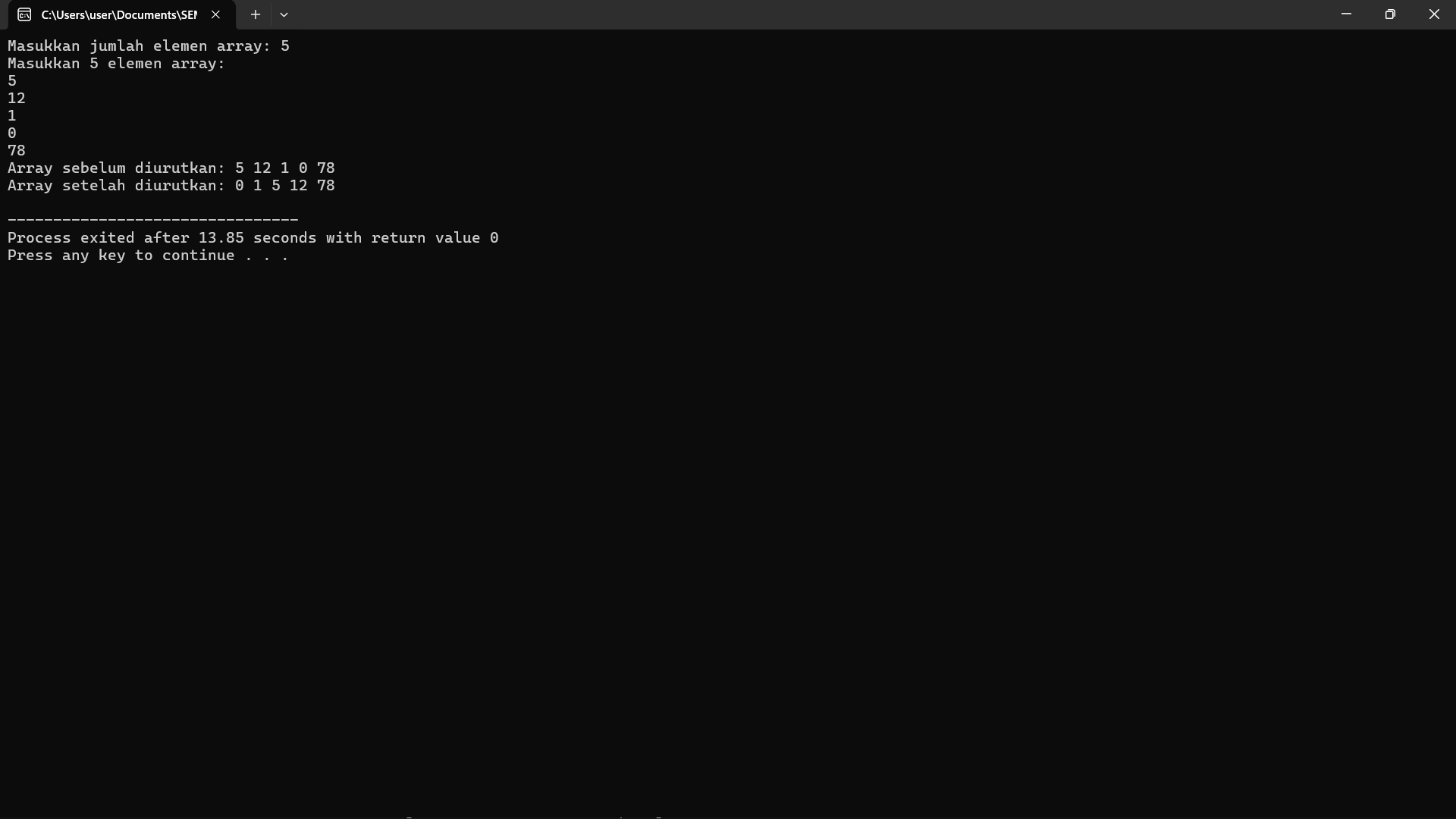
Algoritma Selection Sort sendiri cukup sederhana. Ia mencari elemen terkecil dalam array dan menukarnya dengan elemen pertama. Kemudian, ia mencari elemen terkecil dalam sisa array (yaitu dari elemen kedua hingga terakhir) dan menukarnya dengan elemen kedua, dan seterusnya. Hal ini diulangi hingga seluruh array terurut.

1. **Screenshot program**





1. **Output**



**Merge sort**

1. **Source Code**

//Created by 23343079\_Najwa Eldiara Owilia Tiksa

#include <stdio.h>

// Fungsi untuk menggabungkan dua array terurut menjadi satu array terurut

void merge(int arr[], int l, int m, int r) {

int i, j, k;

int n1 = m - l + 1;

int n2 = r - m;

// Buat array sementara

int L[n1], R[n2];

// Salin data ke array sementara L[] dan R[]

for (i = 0; i < n1; i++)

L[i] = arr[l + i];

for (j = 0; j < n2; j++)

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

// Gabungkan array sementara kembali menjadi arr[l..r]

i = 0;

j = 0;

k = l;

while (i < n1 && j < n2) {

if (L[i] <= R[j]) {

arr[k] = L[i];

i++;

} else {

arr[k] = R[j];

j++;

}

k++;

}

// Salin sisa elemen dari L[], jika ada

while (i < n1) {

arr[k] = L[i];

i++;

k++;

}

// Salin sisa elemen dari R[], jika ada

while (j < n2) {

arr[k] = R[j];

j++;

k++;

}

}

// Fungsi rekursif untuk melakukan pengurutan dengan Merge Sort

void mergeSort(int arr[], int l, int r) {

if (l < r) {

// Temukan titik tengah

int m = l + (r - l) / 2;

// Urutkan setengah bagian kiri dan kanan

mergeSort(arr, l, m);

mergeSort(arr, m + 1, r);

// Gabungkan bagian-bagian yang telah diurutkan

merge(arr, l, m, r);

}

}

// Fungsi untuk mencetak array

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

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

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

printf("\n");

}

int main() {

int arr[] = {6, 4, 16, 10, 3, 25};

int arr\_size = sizeof(arr) / sizeof(arr[0]);

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

printArray(arr, arr\_size);

// Panggil fungsi Merge Sort

mergeSort(arr, 0, arr\_size - 1);

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

printArray(arr, arr\_size);

return 0;

}

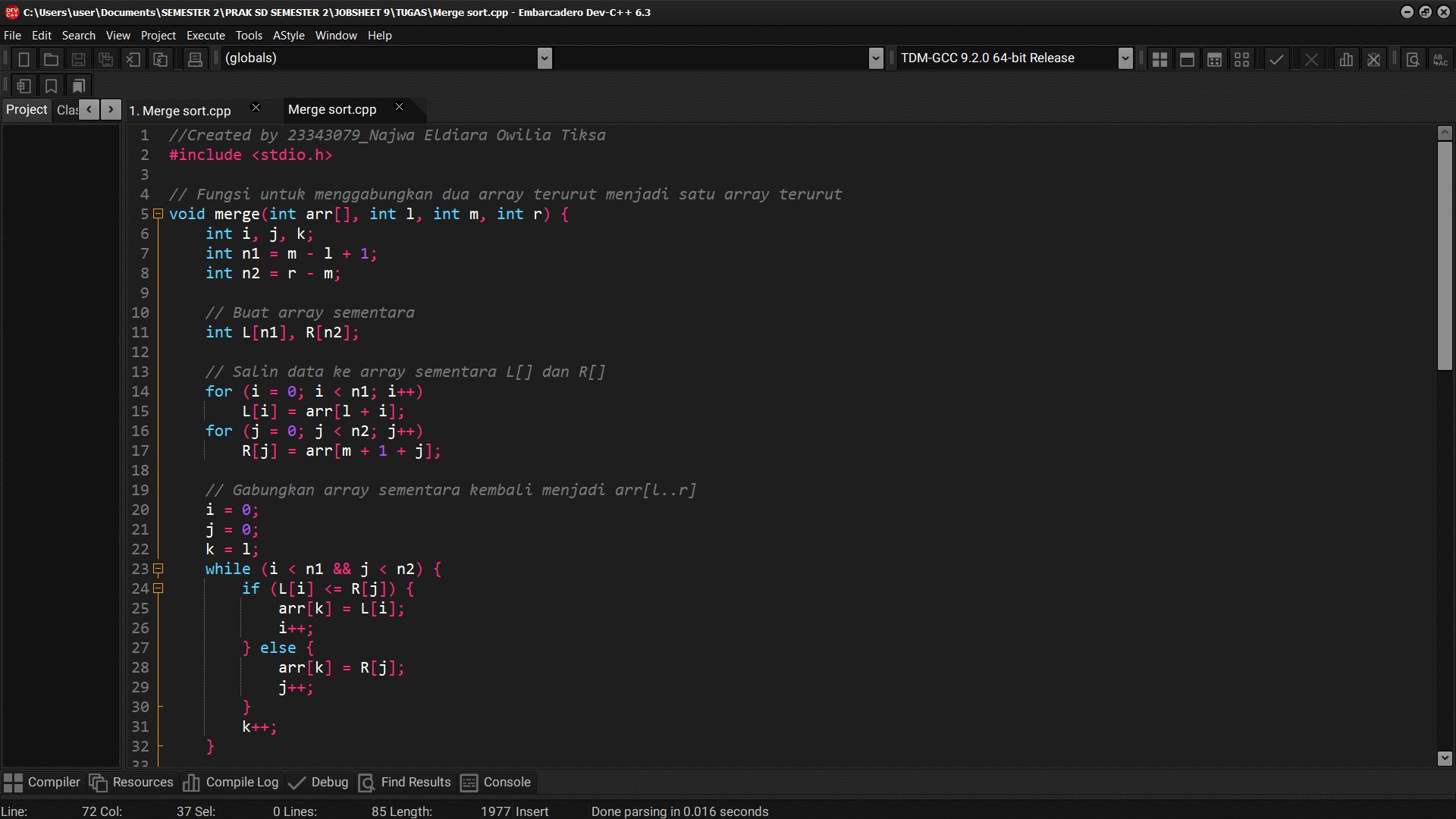
1. **Penjelasan Program**

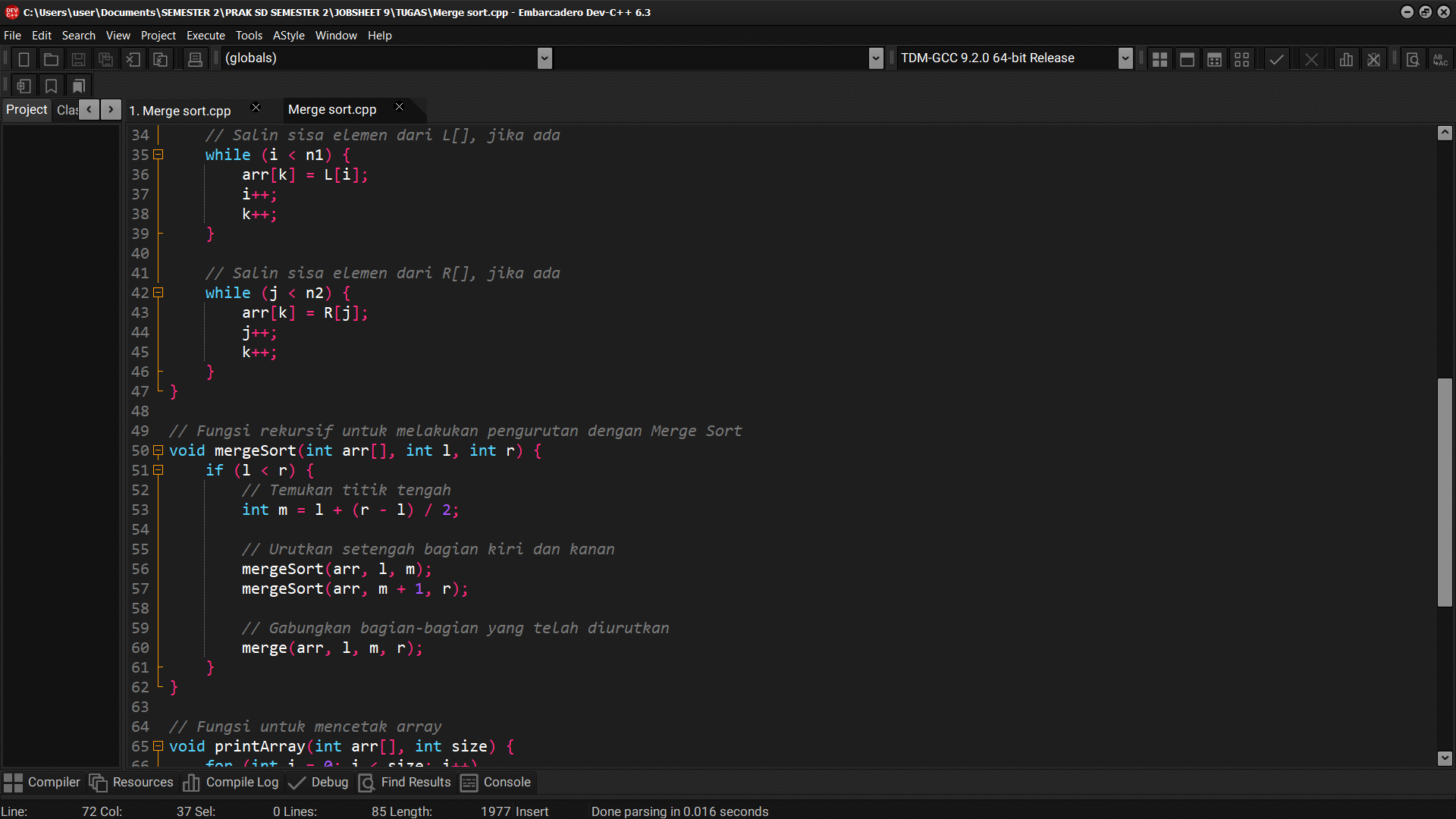
Program ini adalah implementasi dari algoritma Merge Sort dalam bahasa pemrograman C. Merge Sort adalah algoritma pengurutan yang menggunakan pendekatan "divide and conquer" di mana array dibagi menjadi dua bagian, setiap bagian diurutkan secara terpisah, dan kemudian digabungkan kembali secara berurutan.

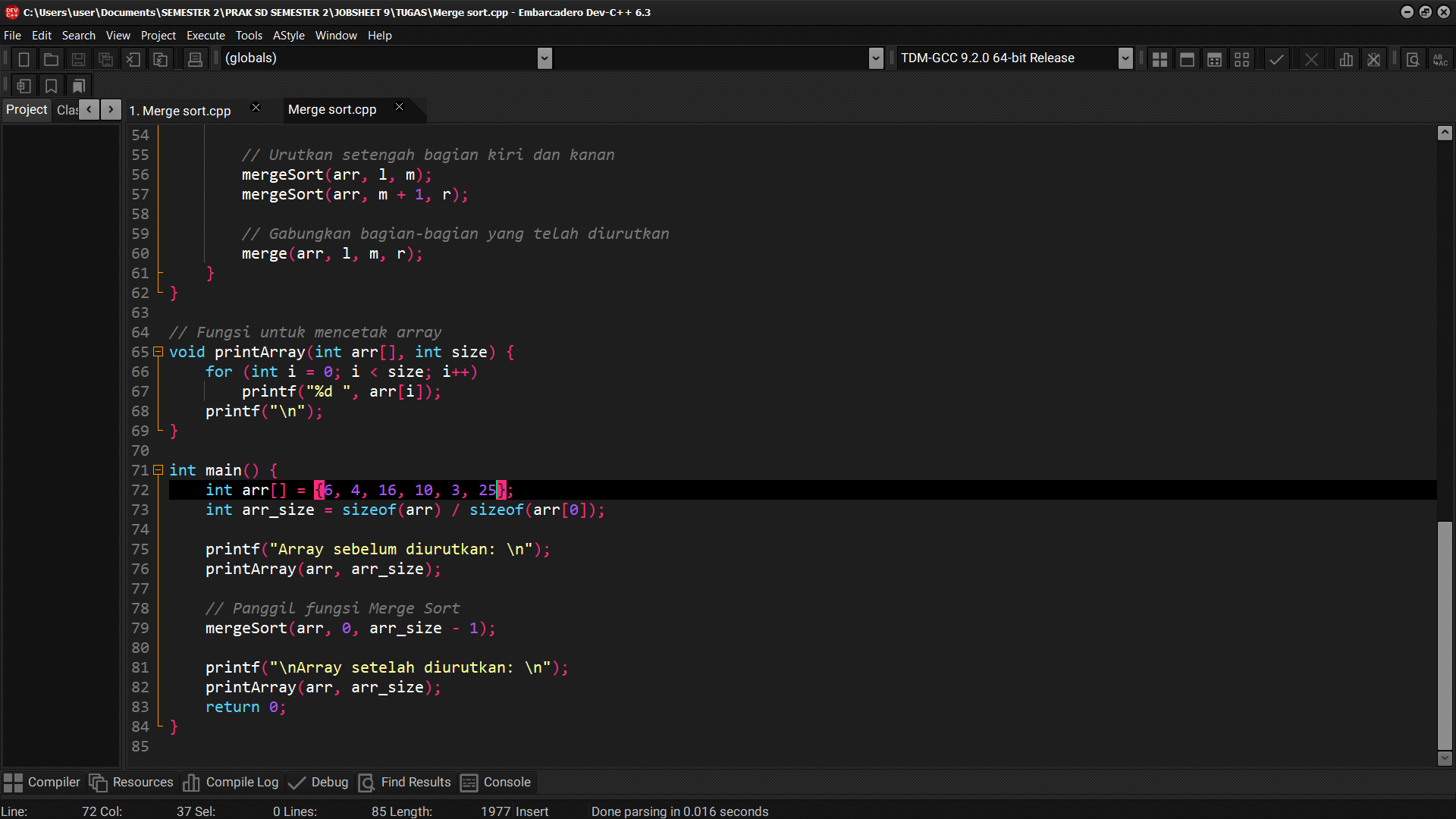
mergeSort: Fungsi rekursif ini melakukan pengurutan dengan Merge Sort. Pertama, fungsi ini memeriksa apakah indeks awal `l` kurang dari indeks akhir `r`, jika tidak, tidak ada yang perlu dilakukan. Jika ya, fungsi ini mencari indeks tengah `m`, kemudian memanggil dirinya sendiri untuk setiap setengah bagian kiri dan kanan dari array, dan terakhir, memanggil fungsi `merge()` untuk menggabungkan kedua bagian yang sudah diurutkan. sebuah array disiapkan dengan elemen-elemen yang akan diurutkan. Array tersebut kemudian dicetak sebelum diurutkan. Selanjutnya, fungsi `mergeSort()` dipanggil untuk mengurutkan array tersebut, dan hasilnya dicetak kembali setelah pengurutan.

Algoritma Merge Sort adalah algoritma dengan kompleksitas waktu O(n log n) di mana n adalah jumlah elemen dalam array. Hal ini membuatnya menjadi salah satu algoritma pengurutan yang efisien.

1. **Screenshot program**







1. **Output**

