**LAPORAN PRATIKUM STRUKTUR DATA**

**JOBSHEET 3**



**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**

**1.**

1. **Source Code**

//created by 23343079\_Najwa Eldiara Owilia Tiksa

#include <stdio.h>

#include <stdlib.h>

void generateFibonacci(int n, int\* fibonacciArray) {

if (n <= 0) {

printf("Masukkan n harus lebih besar dari 0.\n");

return;

}

fibonacciArray[0] = 0;

if (n > 1) {

fibonacciArray[1] = 1;

for (int i = 2; i < n; i++) {

fibonacciArray[i] = fibonacciArray[i - 1] + fibonacciArray[i - 2];

}

}

}

int main() {

int n;

printf("Masukkan nilai n untuk deret Fibonacci: ");

scanf("%d", &n);

if (n <= 0) {

printf("Masukkan n harus lebih besar dari 0.\n");

return 1;

}

// Menggunakan malloc untuk mengalokasikan memori

int\* fibonacciArray = (int\*)malloc(n \* sizeof(int));

// Pengecekan apakah alokasi memori berhasil

if (fibonacciArray == NULL) {

printf("Gagal melakukan alokasi memori.\n");

return 1;

}

// Memanggil fungsi untuk mengisi nilai deret Fibonacci

generateFibonacci(n, fibonacciArray);

// Menampilkan deret Fibonacci

printf("Deret Fibonacci pertama %d angka:\n", n);

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

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

}

printf("\n");

// Menggunakan free untuk membebaskan memori yang dialokasikan

free(fibonacciArray);

return 0;

}

1. **Penjelasan Program**

Program di atas adalah program dalam bahasa C yang bertujuan untuk menghasilkan dan menampilkan deret Fibonacci hingga sejumlah n angka yang dimasukkan oleh pengguna. Mari kita jelaskan bagian-bagian dari program ini dan algoritma yang digunakan:

Algoritma yang digunakan dalam program ini adalah **algoritma iteratif** untuk menghasilkan deret Fibonacci. Algoritma ini bekerja dengan mengisi array Fibonacci dengan dua nilai awal (0 dan 1) dan kemudian menggunakan loop untuk menghitung nilai berikutnya dalam deret dengan menjumlahkan dua nilai sebelumnya. Berikut adalah langkah-langkah algoritma ini:

1. Inisialisasi dua elemen pertama dari deret Fibonacci: fibonacciArray[0] = 0dan fibonacciArray[1] = 1.

2. Gunakan loop untuk menghitung elemen berikutnya dalam deret hingga mencapai nelemen. Untuk setiap elemen ke-i, hitung sebagai jumlah dari dua elemen sebelumnya: fibonacciArray[i] = fibonacciArray[i - 1] + fibonacciArray[i - 2].

Contoh:

Jika pengguna memasukkan nilai n = 1, maka output program akan menjadi:

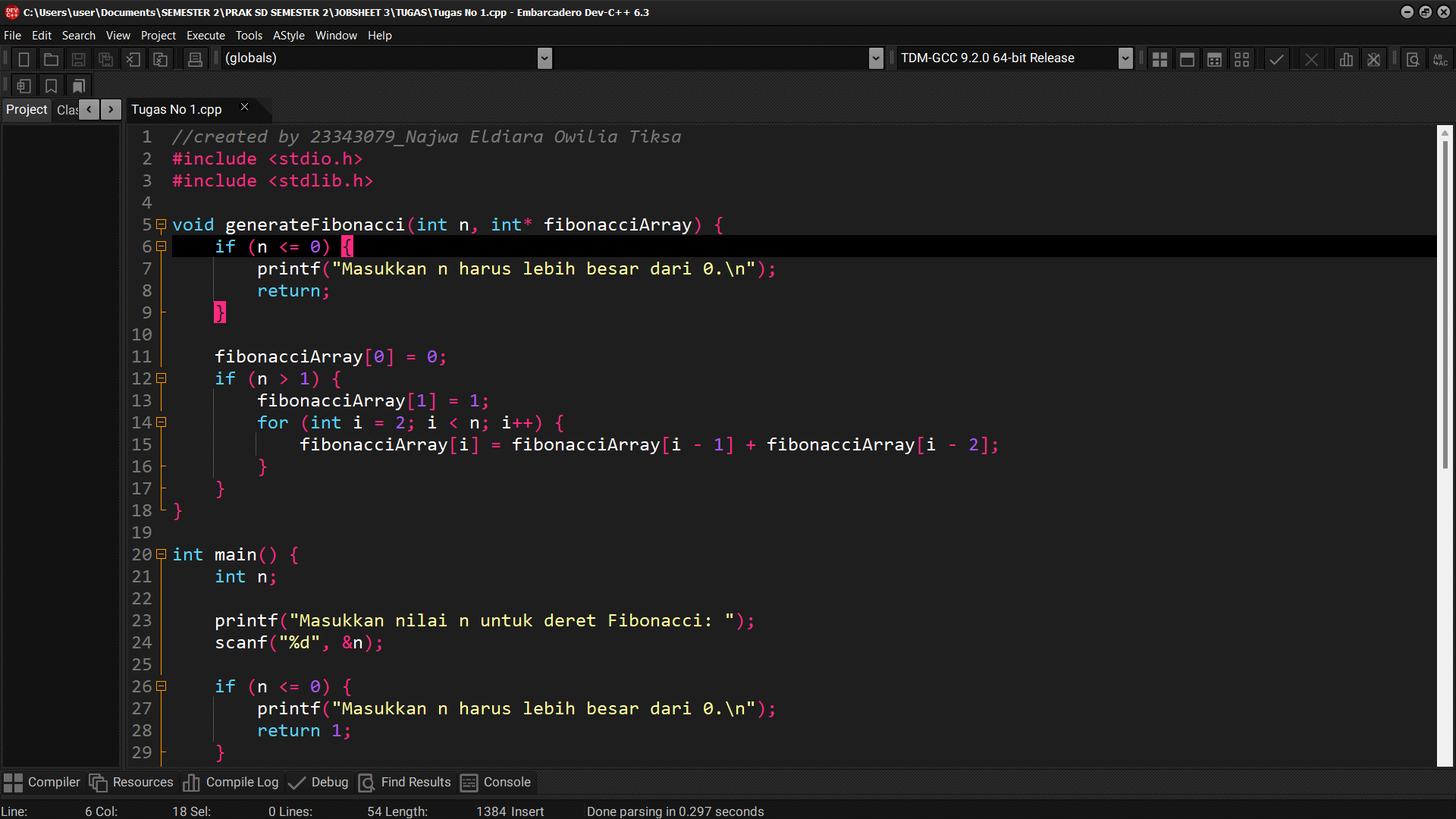
Masukkan nilai n untuk deret Fibonacci: 1

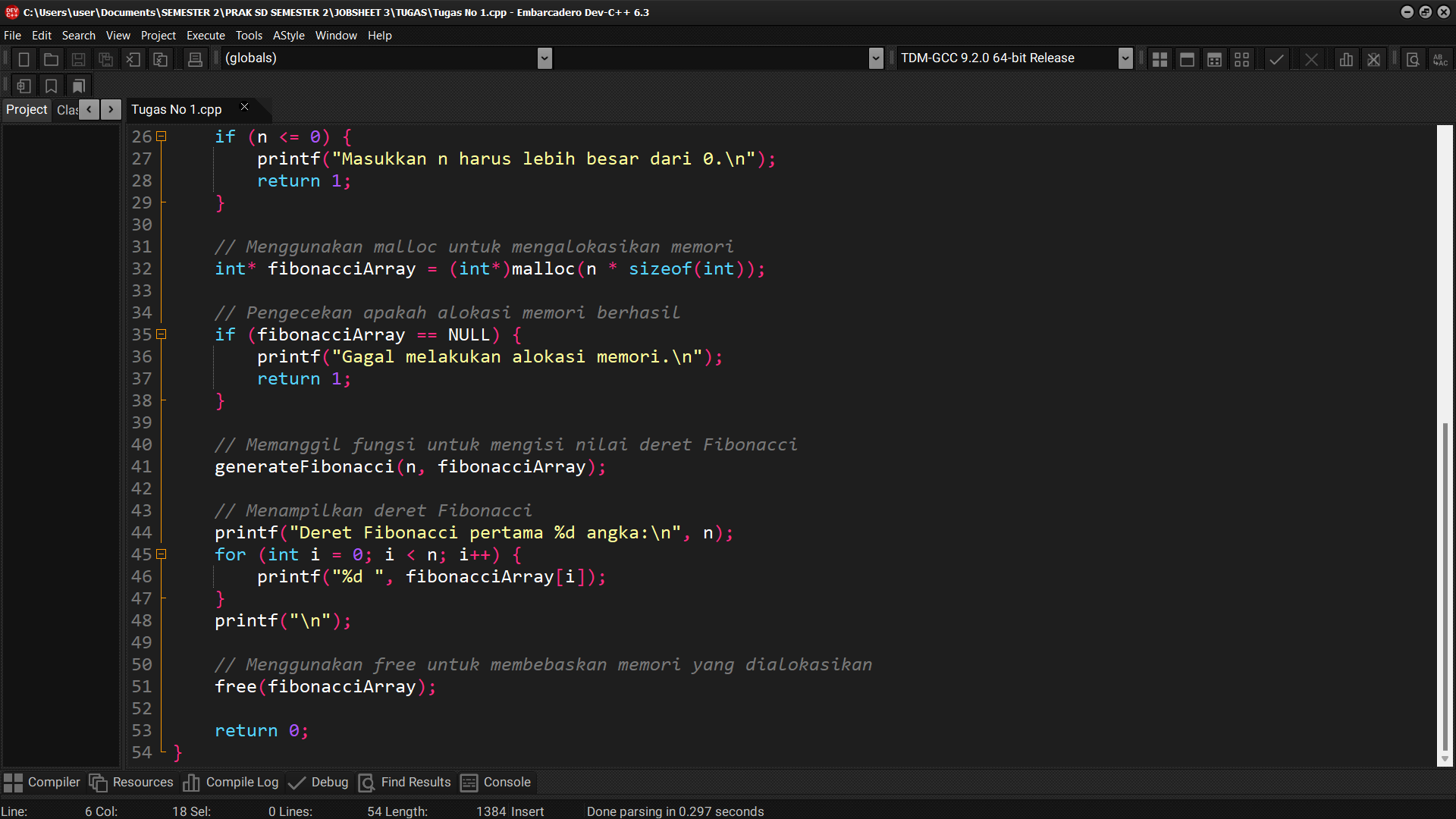
Deret Fibonacci pertama 1 angka:

0

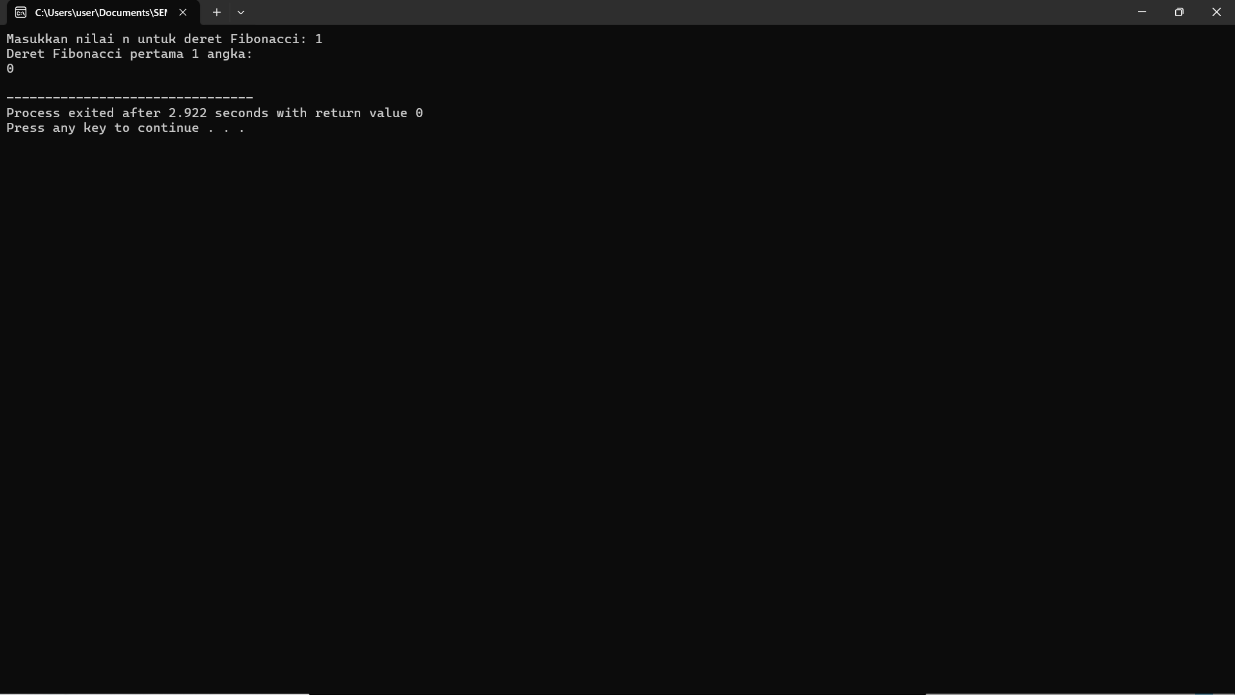
Program ini secara efektif mengalokasikan memori untuk array Fibonacci, mengisi array tersebut dengan nilai-nilai Fibonacci, dan menampilkan deret tersebut. Setelah selesai, program membebaskan memori yang telah dialokasikan.

1. **Screenshot program**





1. **Output**



**2.**

1. **Source Code**

//created by 23343079\_Najwa Eldiara Owilia Tiksa

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

// Fungsi untuk mengecek apakah suatu bilangan adalah prima

bool isPrime(int num) {

if (num < 2) {

return false;

}

for (int i = 2; i \* i <= num; i++) {

if (num % i == 0) {

return false;

}

}

return true;

}

// Fungsi untuk menghasilkan bilangan prima pertama sampai ke-n

void generatePrimes(int n, int\*\* primeArray, int\* size) {

int count = 0;

int\* primes = \*primeArray;

for (int num = 2; count < n; num++) {

if (isPrime(num)) {

// Gunakan realloc untuk mengubah ukuran memori yang dialokasikan

\*primeArray = (int\*)realloc(\*primeArray, (\*size + 1) \* sizeof(int));

if (\*primeArray == NULL) {

printf("Gagal melakukan realokasi memori.\n");

exit(1);

}

primes = \*primeArray;

primes[\*size] = num;

(\*size)++;

count++;

}

}

}

int main() {

int n, m;

int\* primeArray = NULL;

int size = 0;

printf("Masukkan nilai n untuk bilangan prima: ");

scanf("%d", &n);

if (n <= 0) {

printf("Masukkan n harus lebih besar dari 0.\n");

return 1;

}

printf("Masukkan nilai m untuk alokasi awal memori: ");

scanf("%d", &m);

if (m <= 0) {

printf("Masukkan m harus lebih besar dari 0.\n");

return 1;

}

// Menggunakan malloc untuk mengalokasikan memori awal

primeArray = (int\*)malloc(m \* sizeof(int));

// Pengecekan apakah alokasi memori berhasil

if (primeArray == NULL) {

printf("Gagal melakukan alokasi memori.\n");

return 1;

}

// Memanggil fungsi untuk mengisi nilai bilangan prima

generatePrimes(n, &primeArray, &size);

// Menampilkan bilangan prima

printf("Bilangan prima pertama %d:\n", n);

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

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

}

printf("\n");

// Menggunakan free untuk membebaskan memori yang dialokasikan

free(primeArray);

return 0;

}

1. **Penjelasan Program**

Program di atas adalah sebuah program dalam bahasa C yang bertujuan untuk menghasilkan dan menampilkan bilangan prima pertama sebanyak `n` buah. Program ini menggunakan beberapa konsep penting seperti alokasi dan realokasi memori dinamis, serta penggunaan fungsi

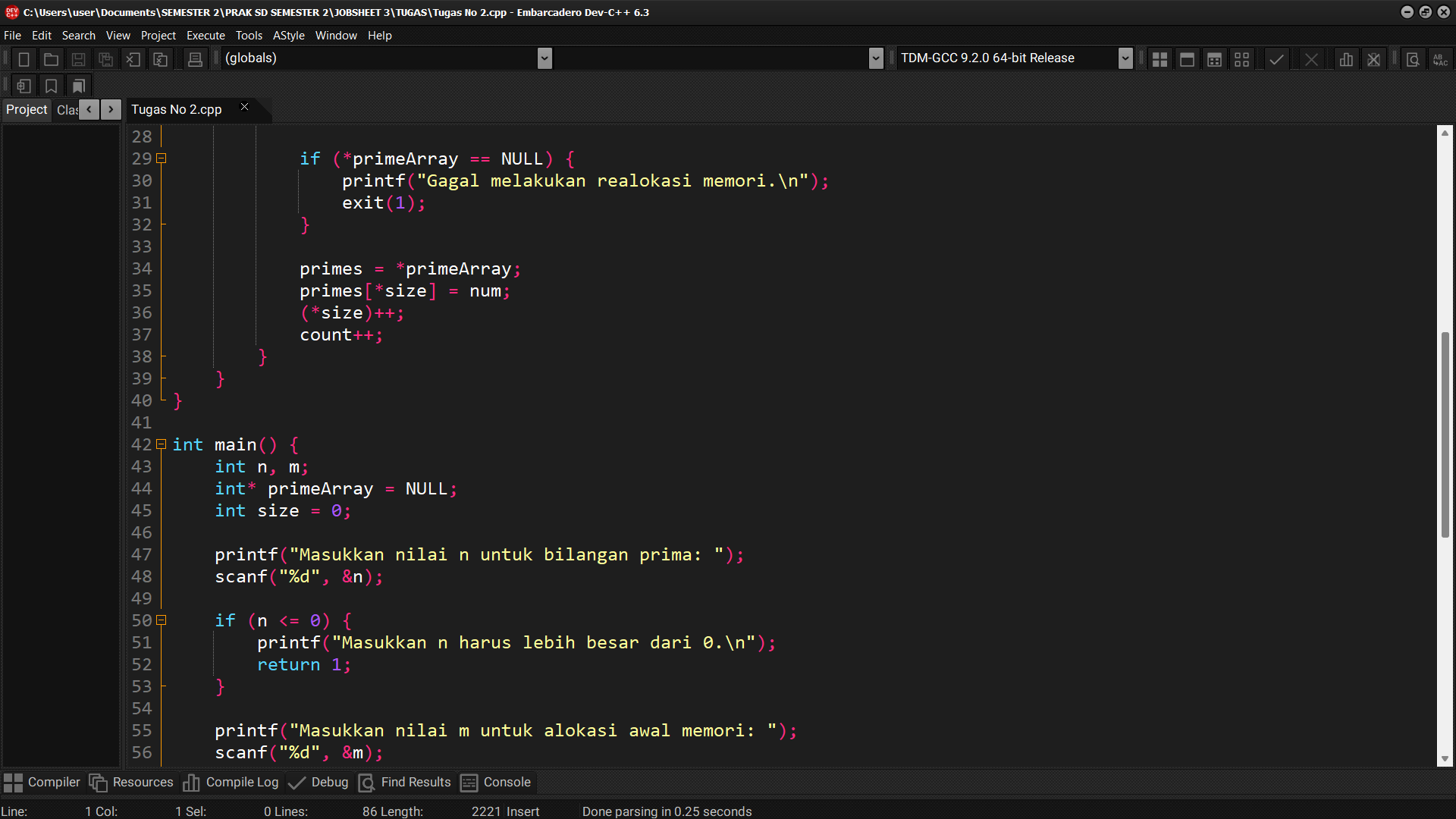
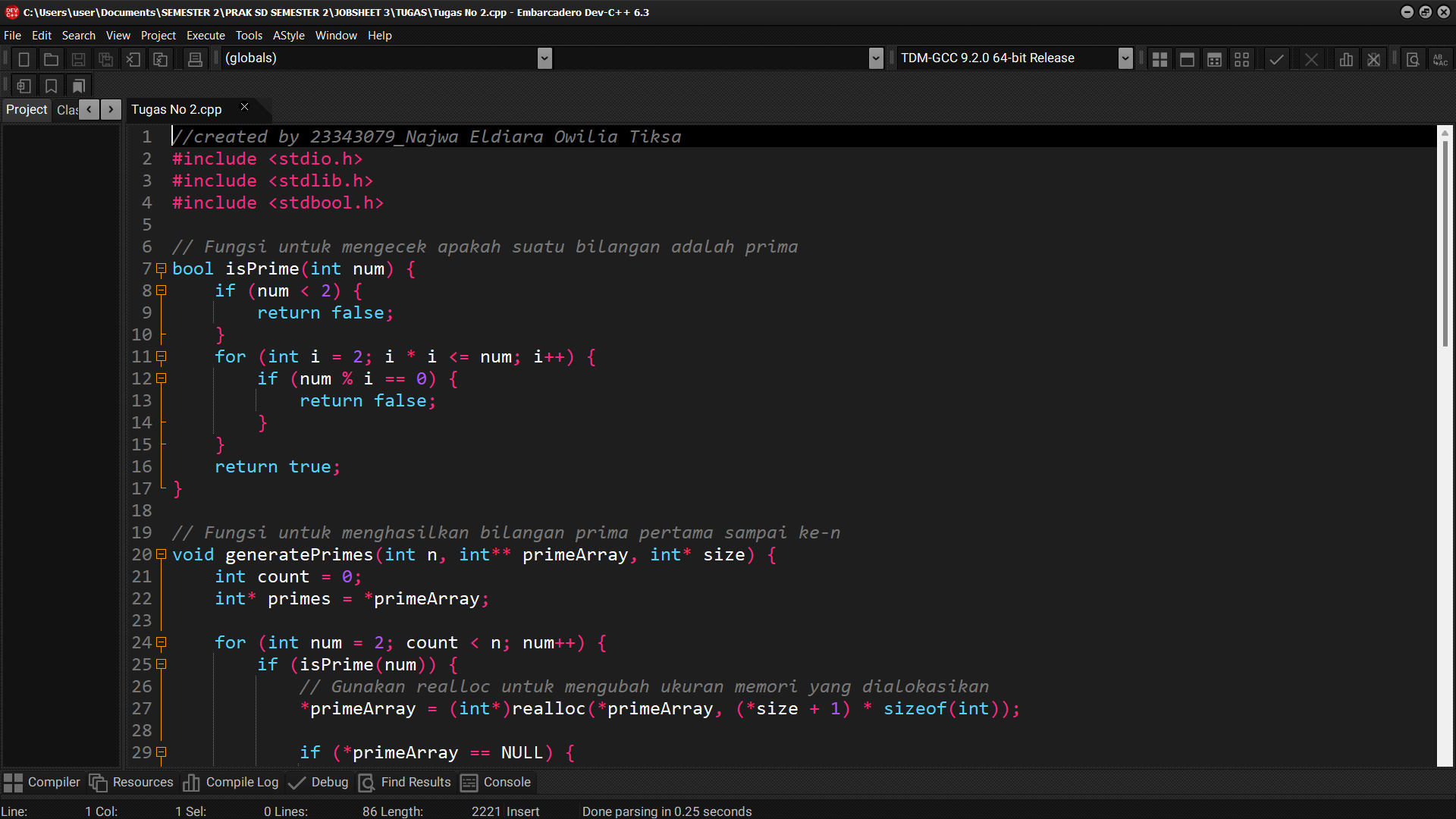
Fungsi ini digunakan untuk mengecek apakah suatu bilangan adalah bilangan prima atau tidak. Algoritmanya adalah dengan memeriksa apakah bilangan tersebut memiliki faktor selain 1 dan dirinya sendiri. Jika ditemukan faktor lain (yaitu, bilangan yang bisa membagi bilangan tersebut tanpa sisa), maka bilangan tersebut bukanlah bilangan prima.

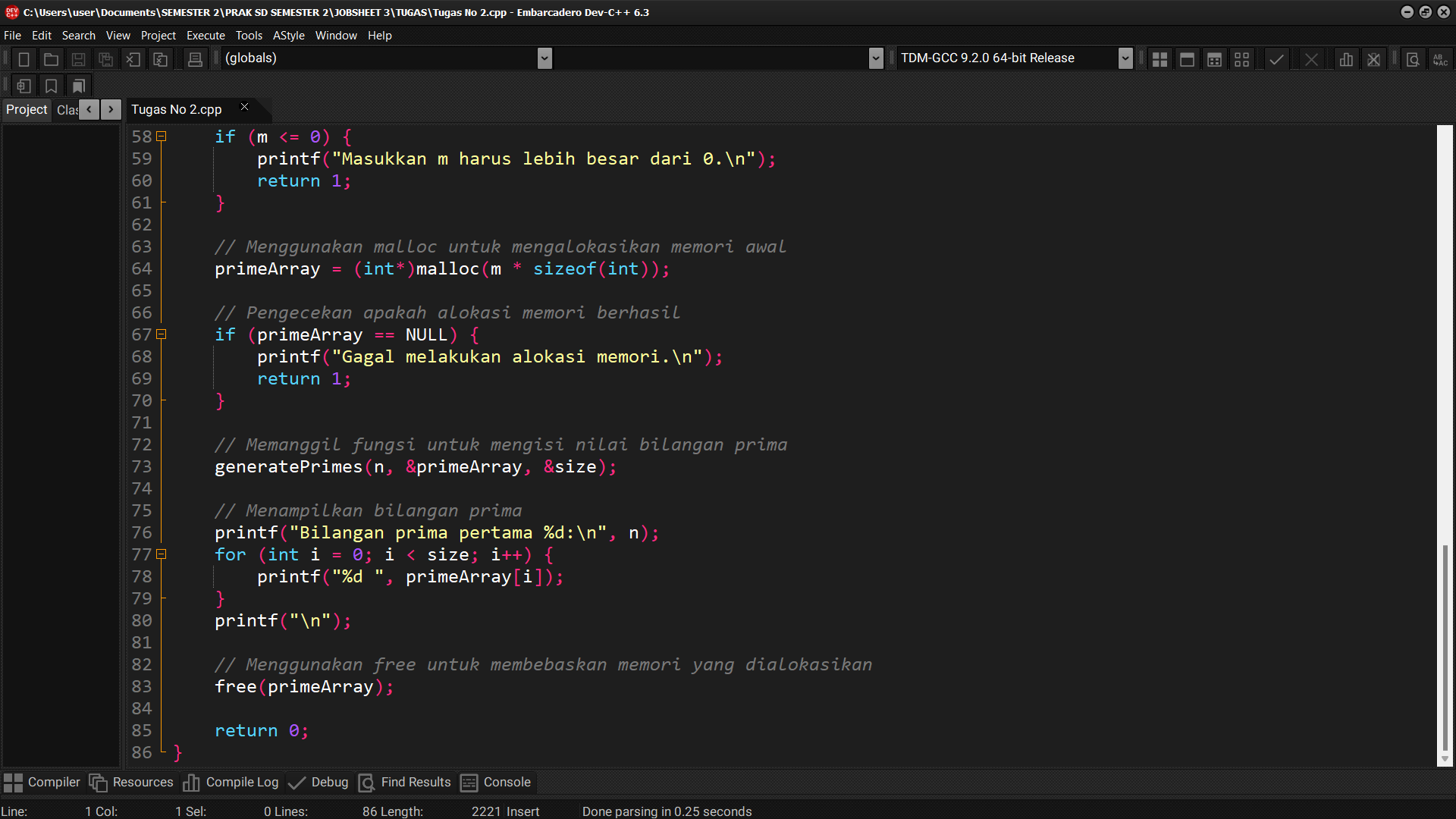
**Algoritma yang digunakan:**

1. **Pengecekan Bilangan Prima (Fungsi `isPrime`)**: Algoritma untuk mengecek bilangan prima ini menggunakan metode trial division hingga akar kuadrat dari bilangan tersebut.
2. **Penggunaan Memori Dinamis (Fungsi `generatePrimes`):** Algoritma ini menggunakan `realloc` untuk menyesuaikan ukuran array dinamis yang menyimpan bilangan prima sesuai kebutuhan.

Secara keseluruhan, program ini menggabungkan konsep dasar dalam pemrograman seperti pengecekan bilangan prima, alokasi memori dinamis, dan manipulasi array untuk mencapai tujuannya menghasilkan dan menampilkan bilangan prima pertama sebanyak `n` buah.

1. **Screenshot program**

****

****

1. **Output**

