**LAPORAN FINAL PROJECT PRAKTIKUM STRUKTUR DATA**

#### “Aplikasi sederhana system manaejemn perpustakaan”



**Disusun Oleh:**

**Serly Eka Putri**

**BP/NIM : 2023/23343083**

|  |  |
| --- | --- |
| **Hari/Tanggal** | **: Selasa / 31-05-2024** |
| **Sesi/Jam** | **: 202323430157/ 08:50 - 10:30 WIB** |

**Dosen Pengampu :**

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

**NIP. 221048**

**FAKULTAS TEKNIK**

#### DEPARTEMEN TEKNIK ELEKTRONIKA

**PROGRAM STUDI S1 TEKNIK INFORMATIKA UNIVERSITAS NEGERI PADANG**

**2024**

**Aplikasi Sederhana Sistem manajemen perpustakaan**

1. **Latar Belakang**

Latar belakng dari program ini adalah kebutuhan untuk mengelola koleksi buku dalam sebuah perpustakaan dengan cara yang efesien dan terorganisir. Program ini dirancang untuk membantu perpustakaan dalam mengelola inventaris buku mereka, memastikan buku-buku dapat dengan mudah ditambahakan, dipinjam, dikembalikan, dan dicari oleh pengguna.

Berikut ini adalah beberapa latar belakang lebih rinci yang mendasari pengembangan program ini :

1. Manajemen Inventaris Buku :

Program ini membantu dalam mencatat dan menyimpa informasi tentang setiap buku, termasuk judul, penulis, dan status peminjamannya.

1. Peminjam dan Pengembalian Buku :

Program ini menyediakan fitur unutuk meminjamndan menegembalikan buku, memastikan bahwa status peminjaman buku selalu terbarui dan akurat.

1. Pencarian Buku :

Program ini memungkinkan pencarian buku berdasarkan judul atau penulis, memudahkan pengguna unutuk menemukan buku yang di mereka cari.

1. Organisasi dan Pengurutan Buku :

Program ini memungkinkan buku diurutkan berdasarkan judul, membuatnya lebih mudah unutuk ditemukan dan diakses.

1. Kemudahan Pengguna :

Program ini dirancang dengan antarmuka yang sederhana dan mudah digunakan, memungkinkan pengguna dari berbagai latar belakang unutuk mengoperasikannya tanpa memerlukan pelatihan teknis yang mendalam.

1. Efesiensi dan Otomatisasi :

Program ini mengurangi bebean kerja staf perpustakkaan dan meningkatkan efesiensi operasional.

Program ini bertujuan untuk meningkatkan efesiensi pengelolan perpustakaan, menyediakan akses yang lebih baik kepada pengguna, dan memastikan bahwa koleksi buku dapat diakses dan dikelola dengan cara yang sistematis dan terorganisir.

1. Source Code

// Created\_by\_Serly Eka Putri\_23343083

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_JUDUL\_LENGTH 50

#define MAX\_PENULIS\_LENGTH 50

typedef struct Buku {

int id;

char judul[MAX\_JUDUL\_LENGTH];

char penulis[MAX\_PENULIS\_LENGTH];

int dipinjam; // 0: tersedia, 1: dipinjam

struct Buku \*next;

} Buku;

Buku \*head = NULL;

int jumlahBuku = 0;

void tambahBuku() {

Buku \*bukuBaru = (Buku \*)malloc(sizeof(Buku));

if (!bukuBaru) {

printf("Gagal mengalokasikan memori.\n");

return;

}

bukuBaru->id = jumlahBuku + 1;

printf("Masukkan judul: ");

scanf(" %[^\n]%\*c", bukuBaru->judul); // Membaca string dengan spasi

printf("Masukkan penulis: ");

scanf(" %[^\n]%\*c", bukuBaru->penulis);

bukuBaru->dipinjam = 0;

bukuBaru->next = NULL;

if (!head) {

head = bukuBaru;

} else {

Buku \*temp = head;

while (temp->next) {

temp = temp->next;

}

temp->next = bukuBaru;

}

jumlahBuku++;

printf("Buku berhasil ditambahkan!\n");

}

void pinjamBuku() {

int id;

printf("Masukkan ID buku yang ingin dipinjam: ");

scanf("%d", &id);

Buku \*temp = head;

while (temp) {

if (temp->id == id) {

if (temp->dipinjam) {

printf("Buku sudah dipinjam.\n");

} else {

temp->dipinjam = 1;

printf("Anda berhasil meminjam \"%s\".\n", temp->judul);

}

return;

}

temp = temp->next;

}

printf("Buku tidak ditemukan.\n");

}

void kembalikanBuku() {

int id;

printf("Masukkan ID buku yang ingin dikembalikan: ");

scanf("%d", &id);

Buku \*temp = head;

while (temp) {

if (temp->id == id) {

if (!temp->dipinjam) {

printf("Buku tidak dipinjam.\n");

} else {

temp->dipinjam = 0;

printf("Anda berhasil mengembalikan \"%s\".\n", temp->judul);

}

return;

}

temp = temp->next;

}

printf("Buku tidak ditemukan.\n");

}

void cariBuku() {

char kataKunci[MAX\_JUDUL\_LENGTH];

printf("Masukkan judul atau penulis untuk mencari: ");

scanf(" %[^\n]%\*c", kataKunci);

Buku \*temp = head;

while (temp) {

if (strstr(temp->judul, kataKunci) != NULL || strstr(temp->penulis, kataKunci) != NULL) {

printf("ID: %d, Judul: %s, Penulis: %s, %s\n", temp->id, temp->judul, temp->penulis,

temp->dipinjam ? "Dipinjam" : "Tersedia");

}

temp = temp->next;

}

}

void daftarBuku() {

printf("\nDaftar buku di perpustakaan:\n");

Buku \*temp = head;

while (temp) {

printf("ID: %d, Judul: %s, Penulis: %s, %s\n", temp->id, temp->judul, temp->penulis,

temp->dipinjam ? "Dipinjam" : "Tersedia");

temp = temp->next;

}

}

void tukarBuku(Buku \*a, Buku \*b) {

int idTemp = a->id;

char judulTemp[MAX\_JUDUL\_LENGTH];

char penulisTemp[MAX\_PENULIS\_LENGTH];

int dipinjamTemp = a->dipinjam;

strcpy(judulTemp, a->judul);

strcpy(penulisTemp, a->penulis);

a->id = b->id;

strcpy(a->judul, b->judul);

strcpy(a->penulis, b->penulis);

a->dipinjam = b->dipinjam;

b->id = idTemp;

strcpy(b->judul, judulTemp);

strcpy(b->penulis, penulisTemp);

b->dipinjam = dipinjamTemp;

}

void sortBuku() {

if (!head || !head->next) {

return;

}

int swapped;

Buku \*ptr1;

Buku \*lptr = NULL;

do {

swapped = 0;

ptr1 = head;

while (ptr1->next != lptr) {

if (strcmp(ptr1->judul, ptr1->next->judul) > 0) {

tukarBuku(ptr1, ptr1->next);

swapped = 1;

}

ptr1 = ptr1->next;

}

lptr = ptr1;

} while (swapped);

printf("Buku berhasil diurutkan berdasarkan judul.\n");

}

int main() {

int pilihan;

while (1) {

printf("\nSistem Manajemen Perpustakaan\n");

printf("1. Tambah Buku\n");

printf("2. Pinjam Buku\n");

printf("3. Kembalikan Buku\n");

printf("4. Cari Buku\n");

printf("5. Daftar Semua Buku\n");

printf("6. Urutkan Buku Berdasarkan Judul\n");

printf("7. Keluar\n");

printf("Masukkan pilihan Anda: ");

scanf("%d", &pilihan);

switch (pilihan) {

case 1:

tambahBuku();

break;

case 2:

pinjamBuku();

break;

case 3:

kembalikanBuku();

break;

case 4:

cariBuku();

break;

case 5:

daftarBuku();

break;

case 6:

sortBuku();

break;

case 7:

exit(0);

default:

printf("Pilihan tidak valid!\n");

}

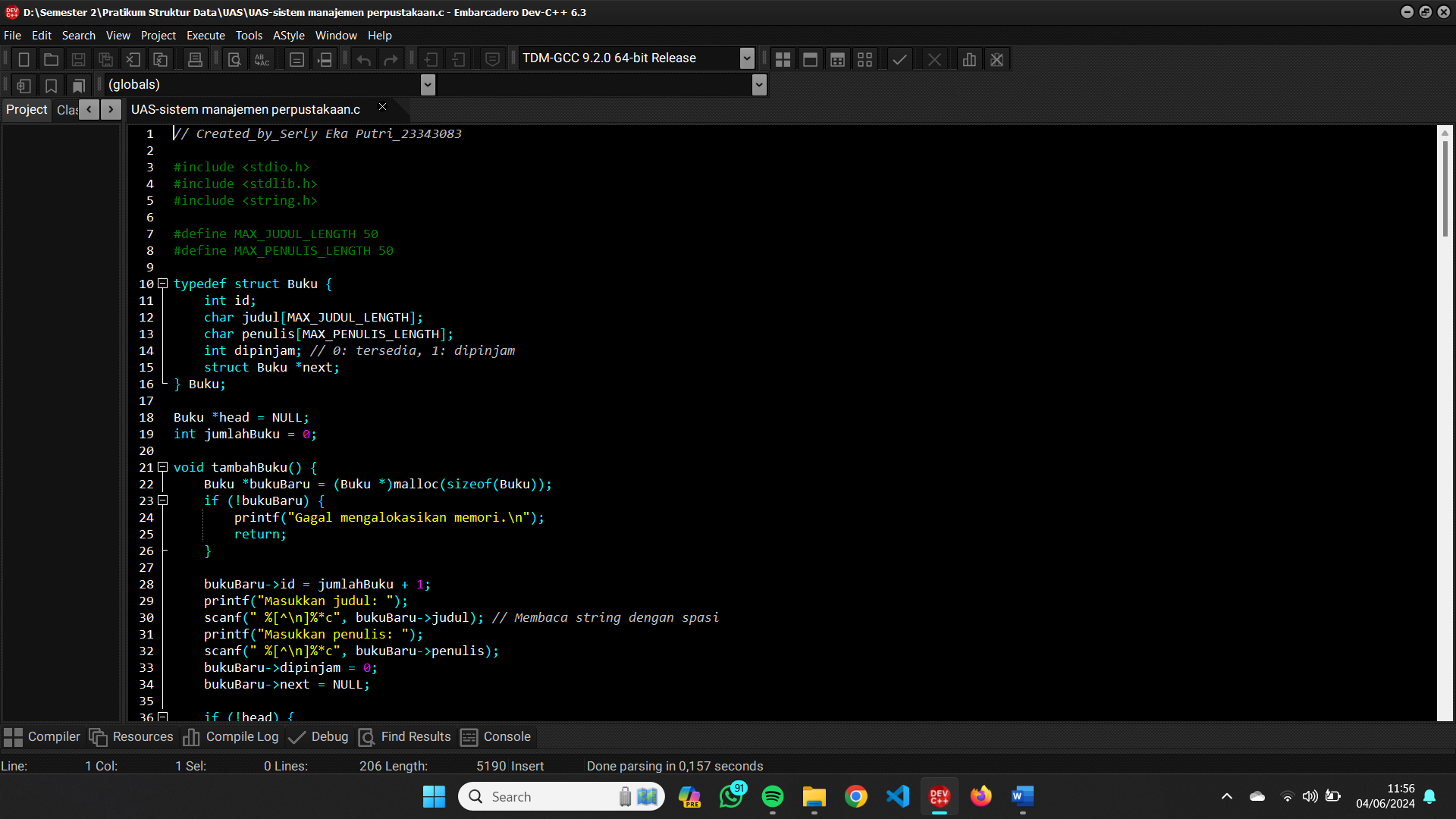
}

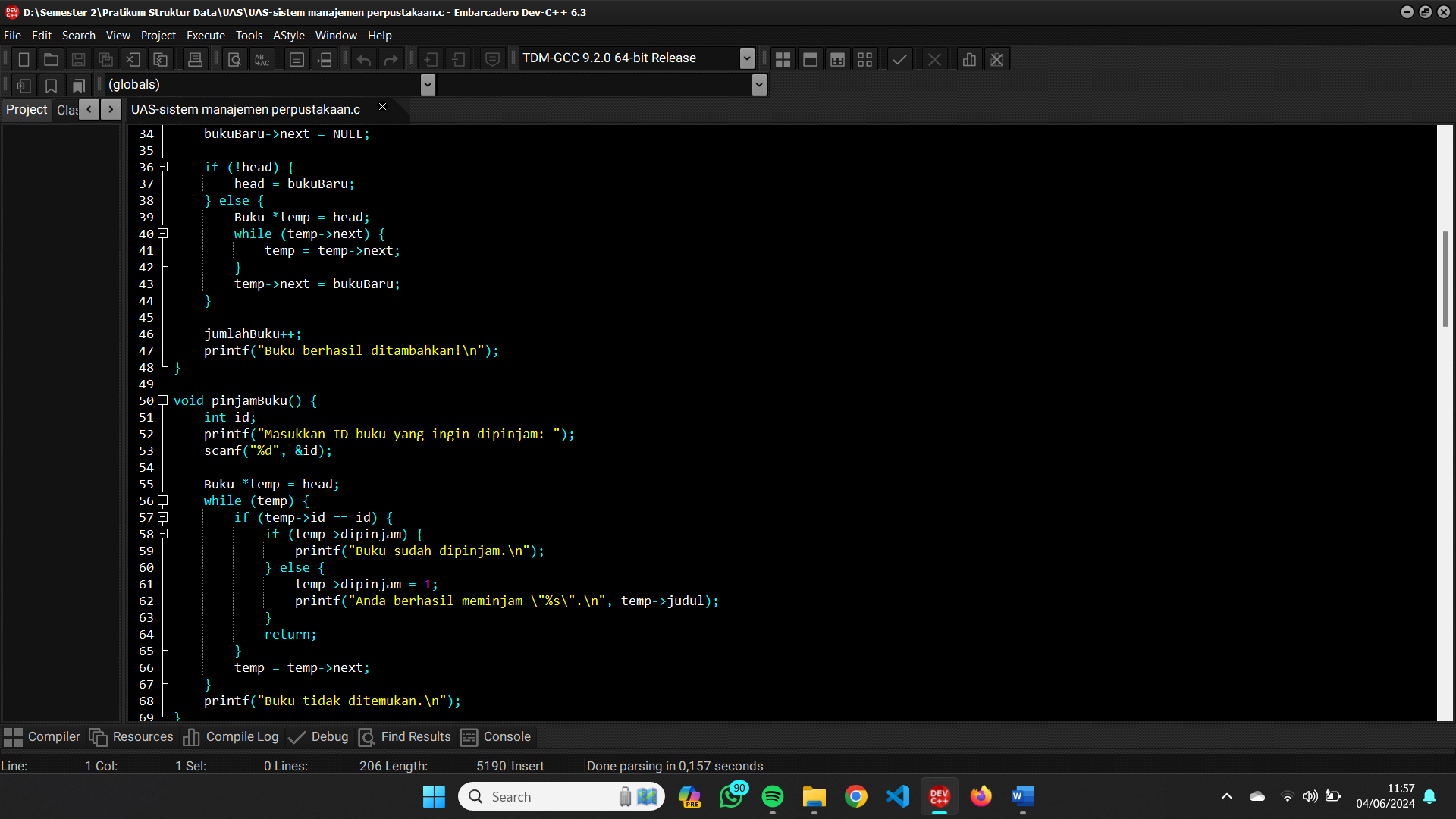
return 0;

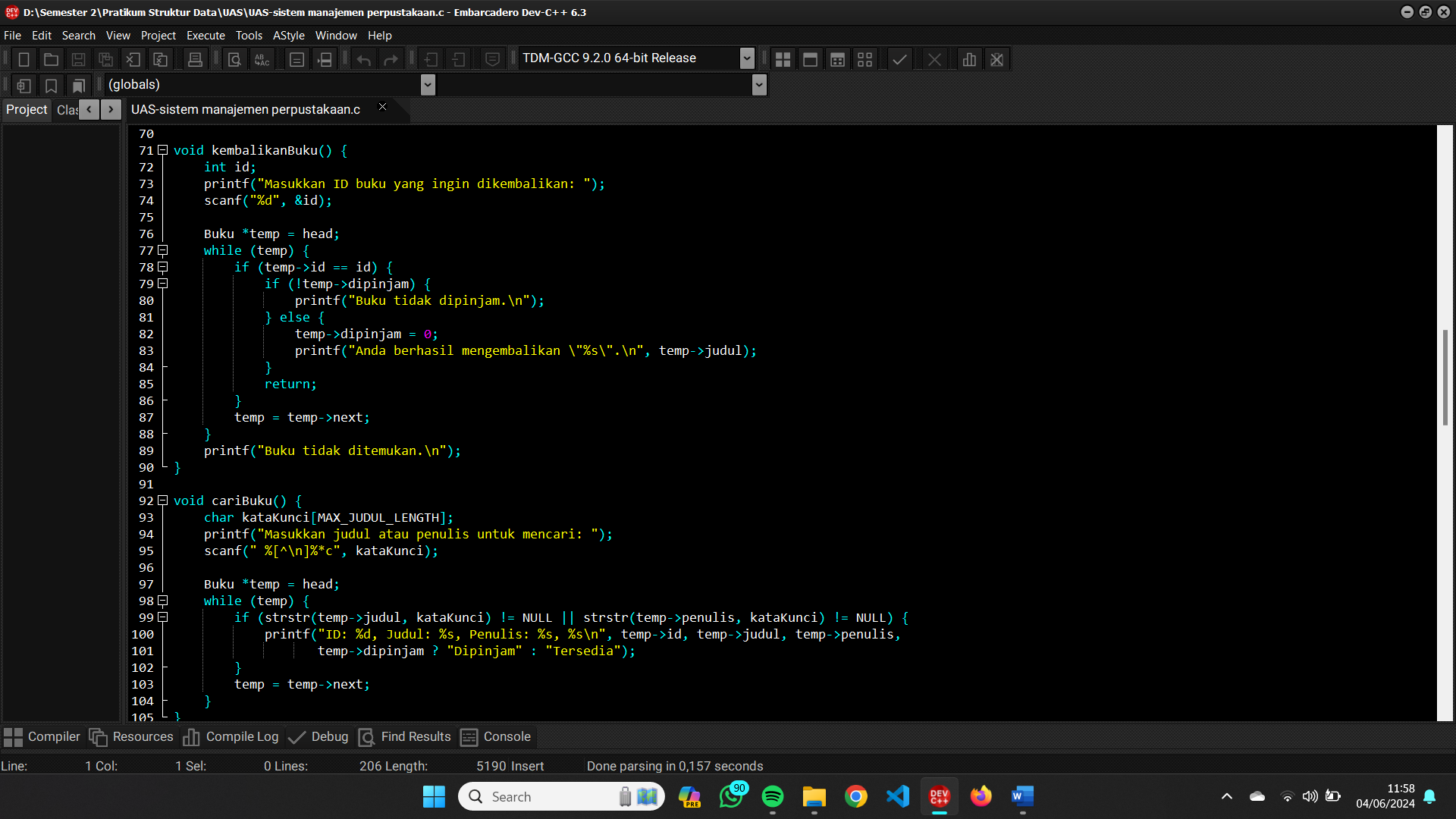
}

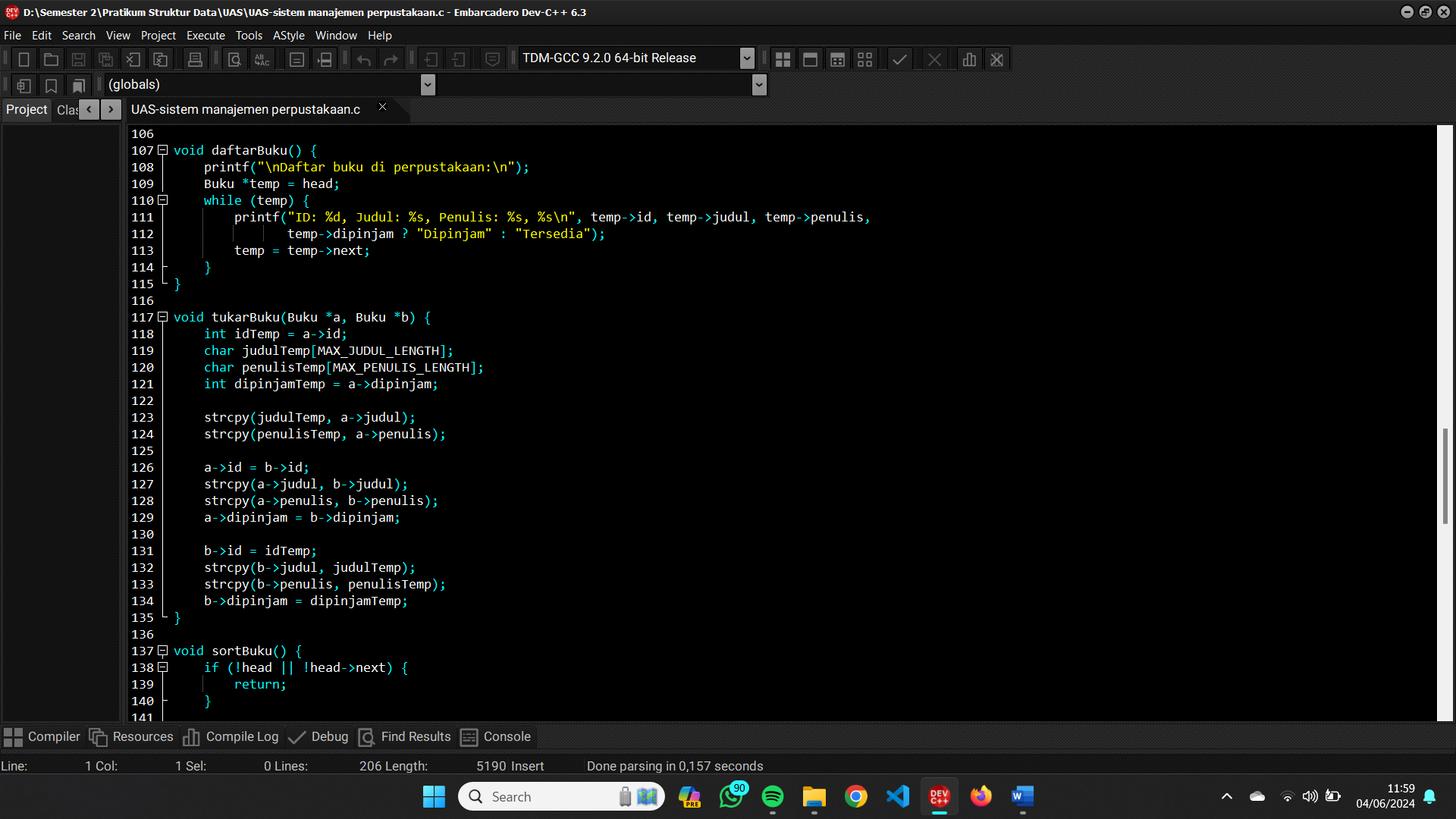
1. Hasilnya / Screenshot

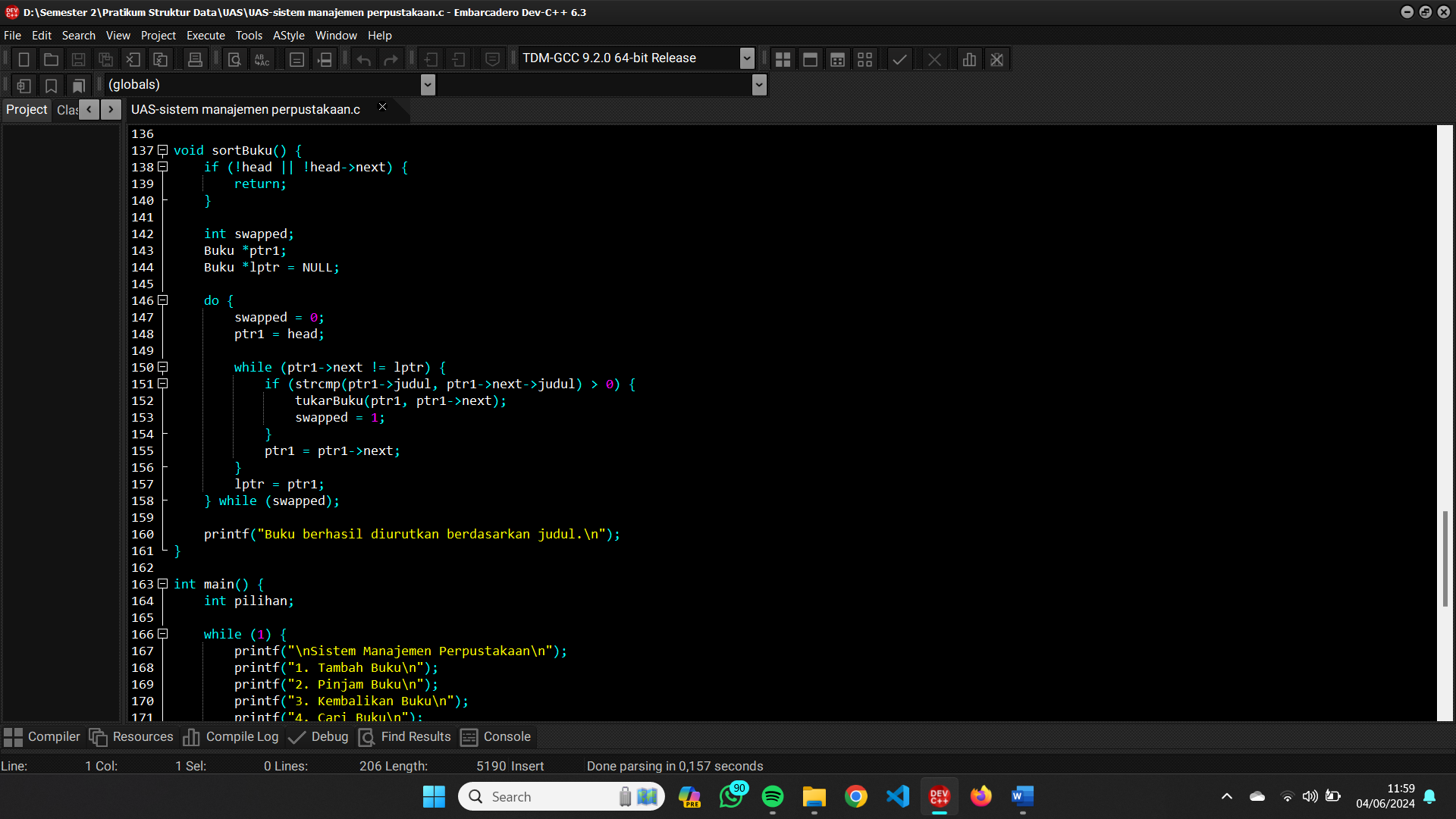
Screenshot Code

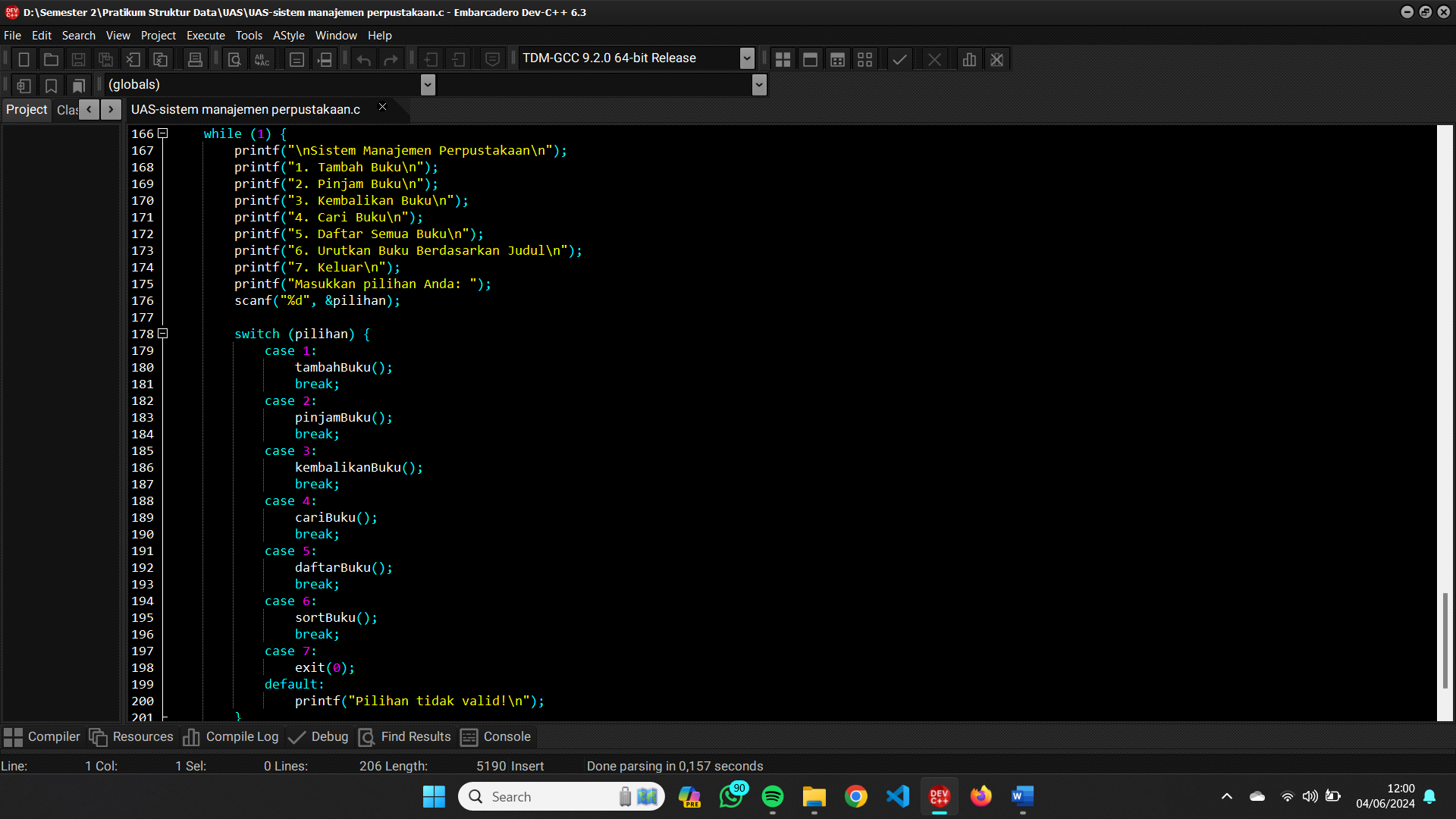


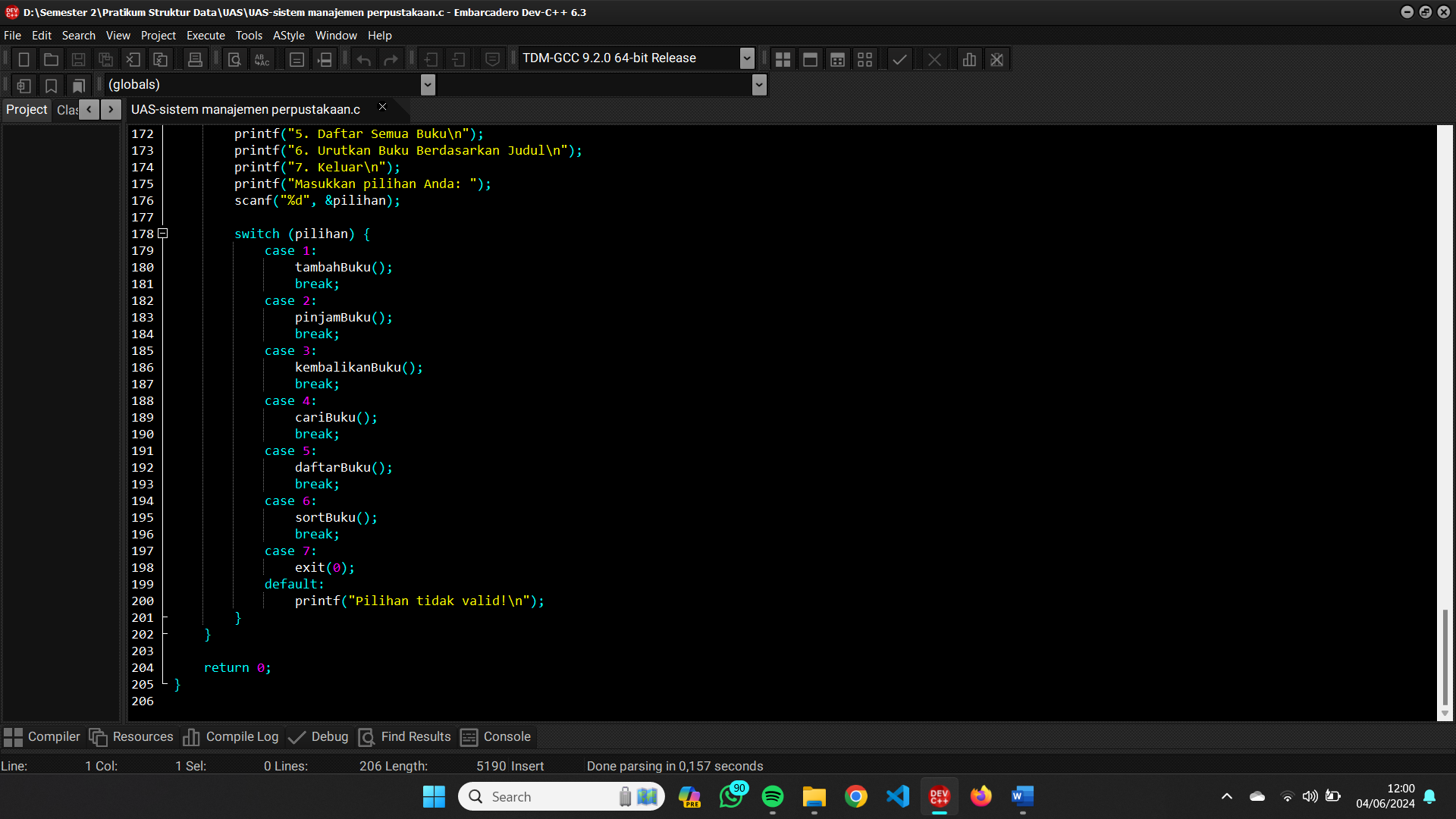




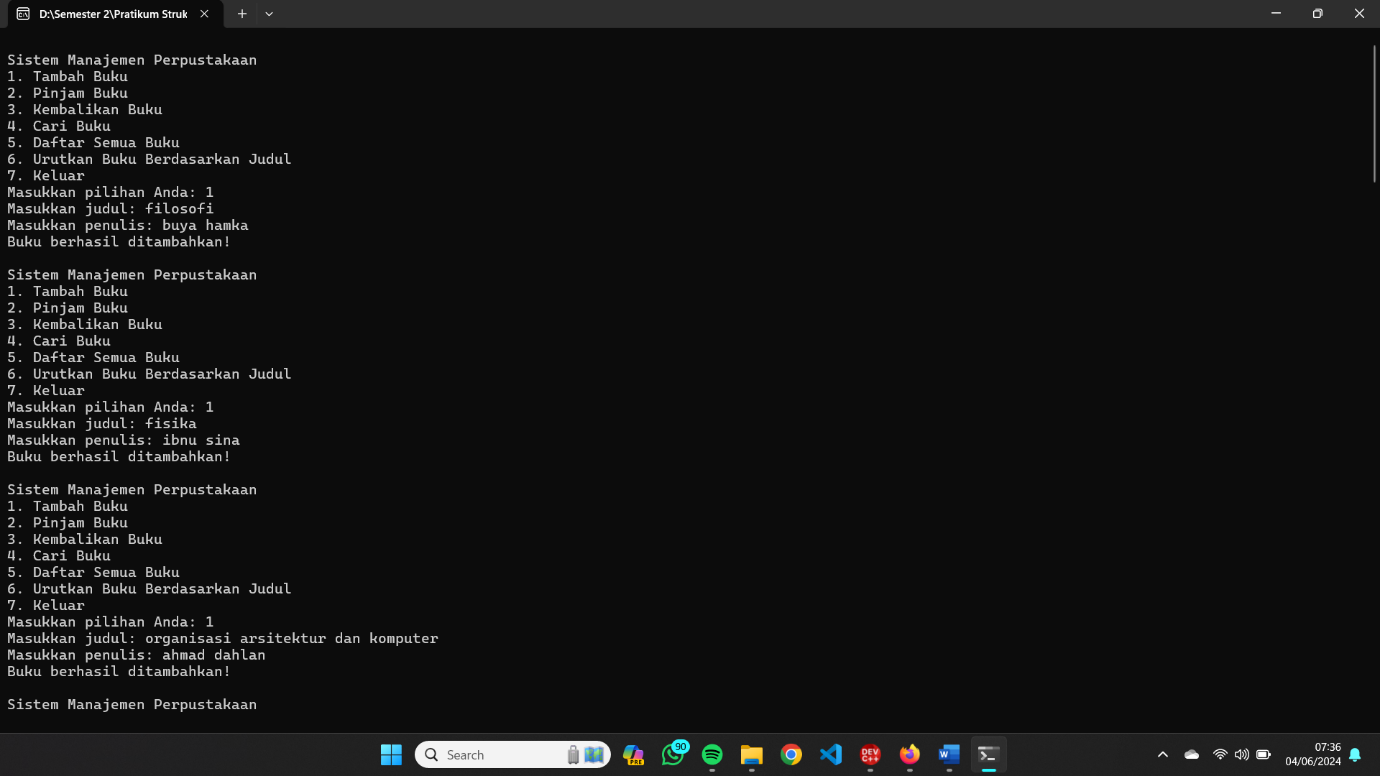


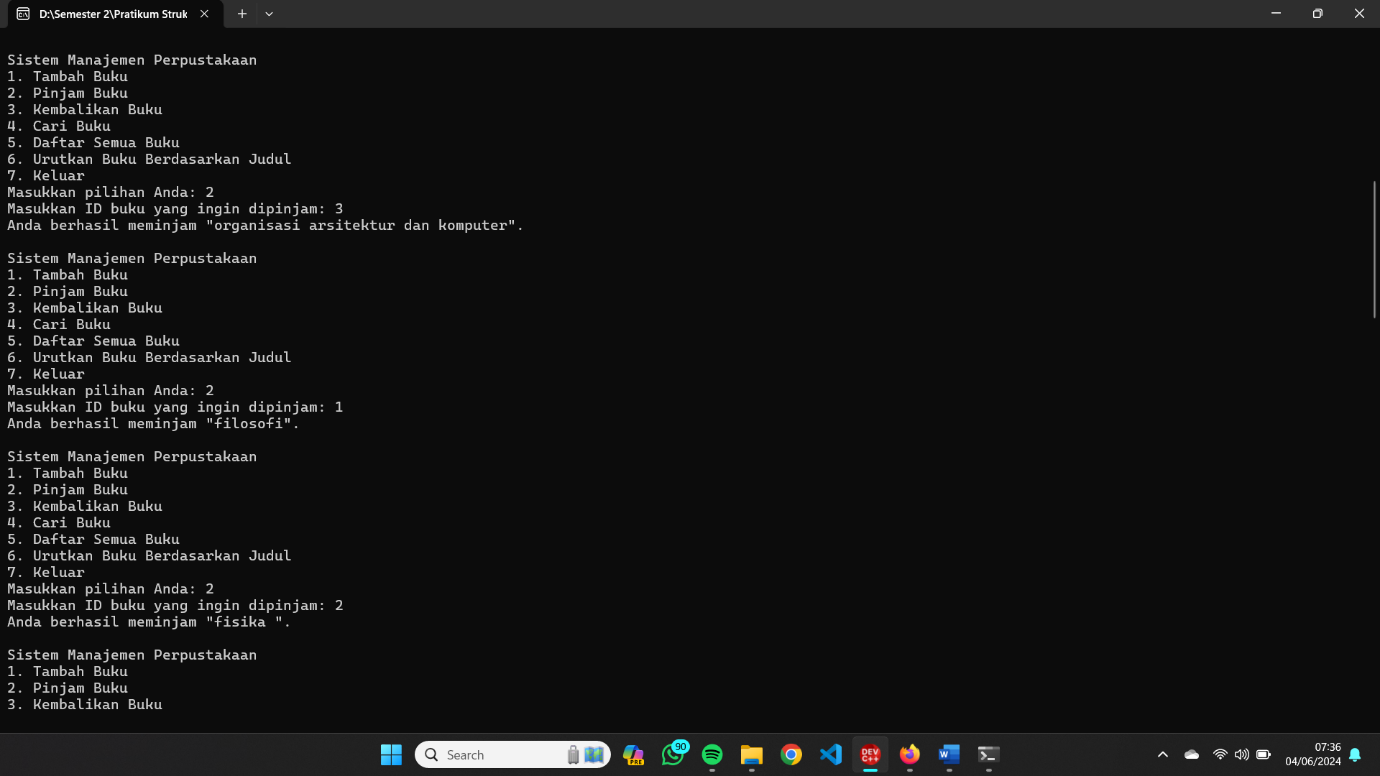


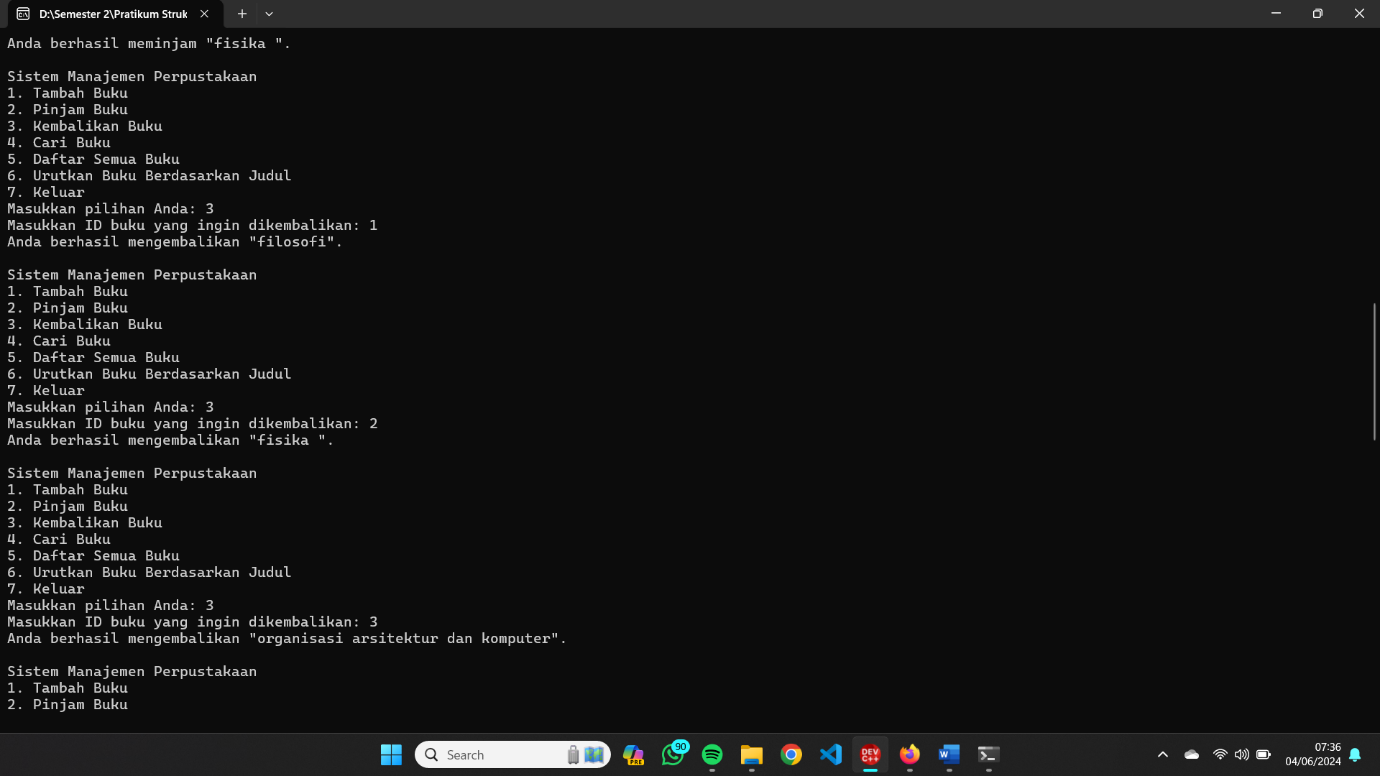


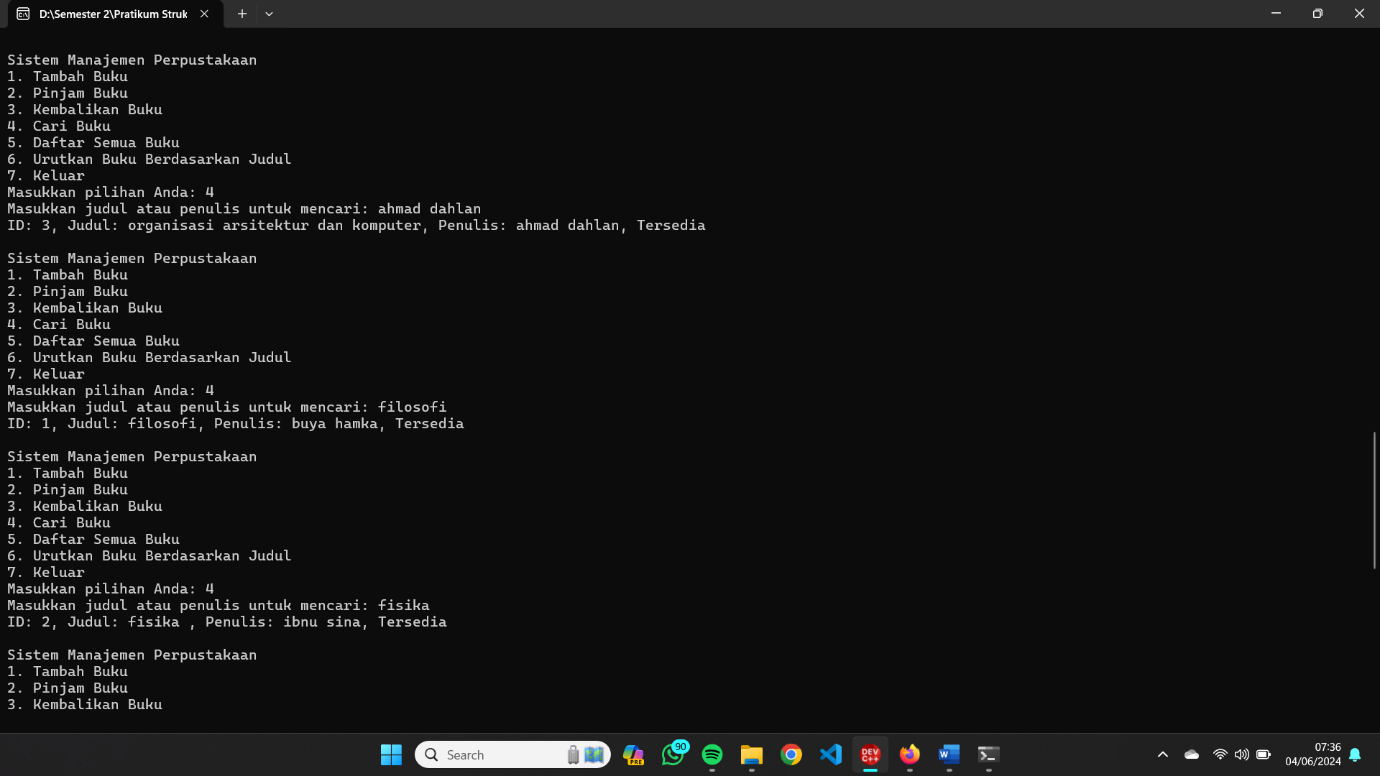


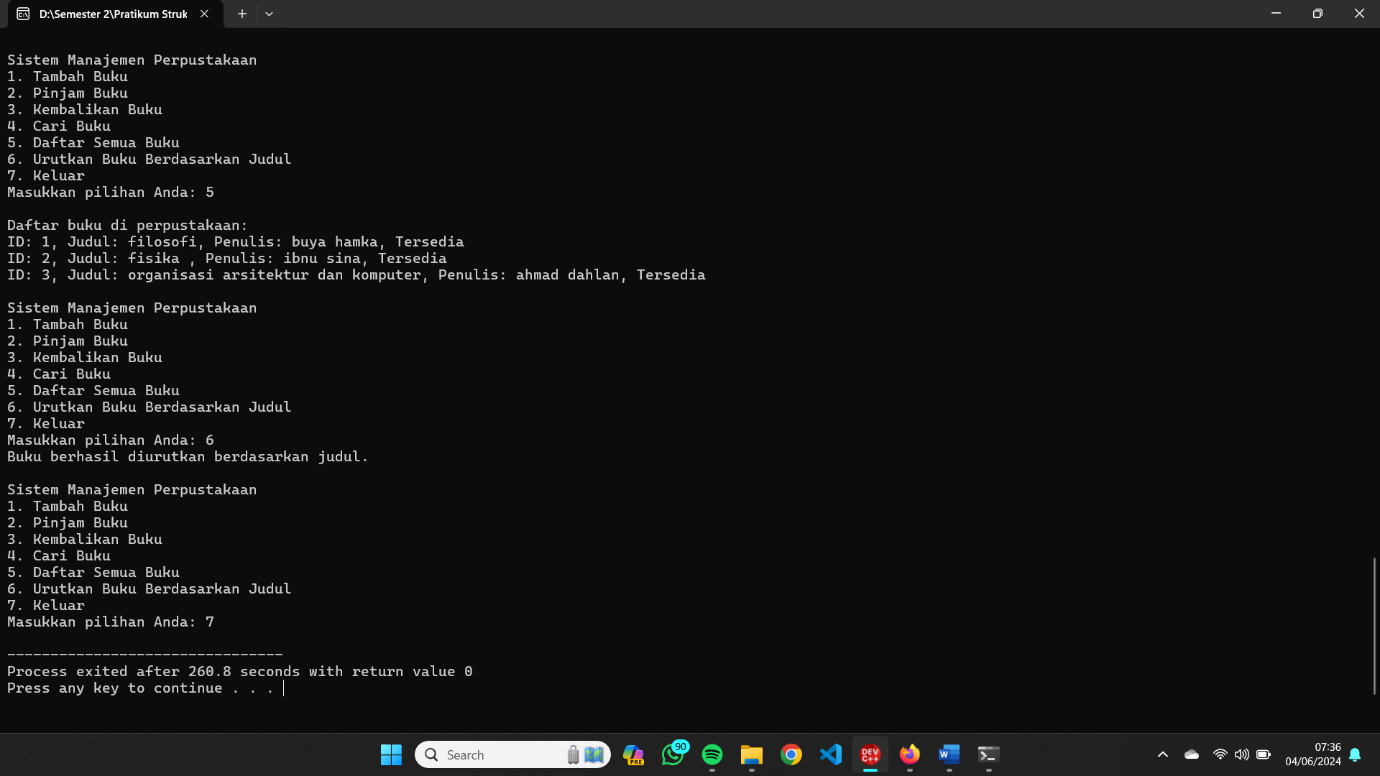
Screenshoot hasil



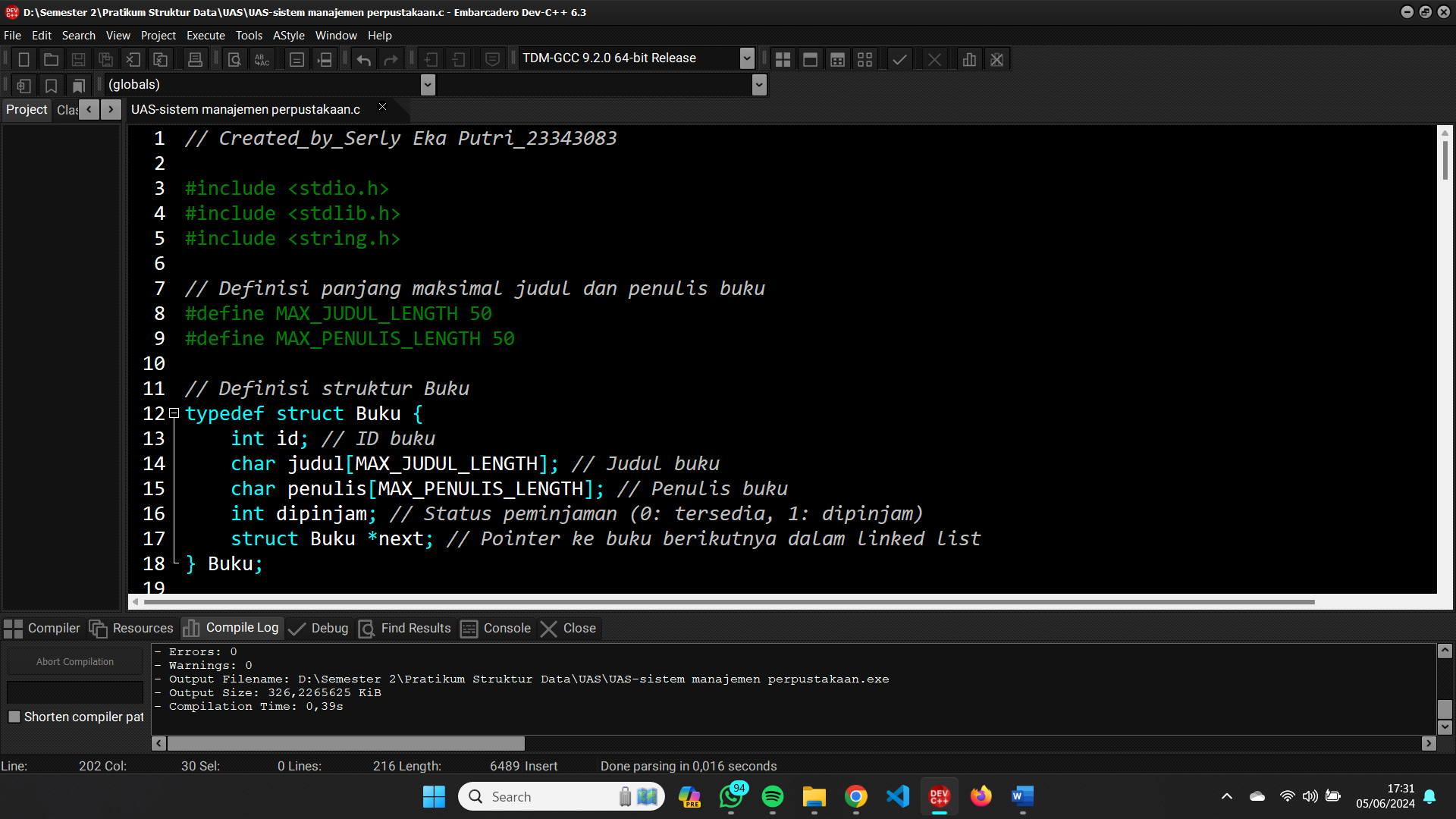




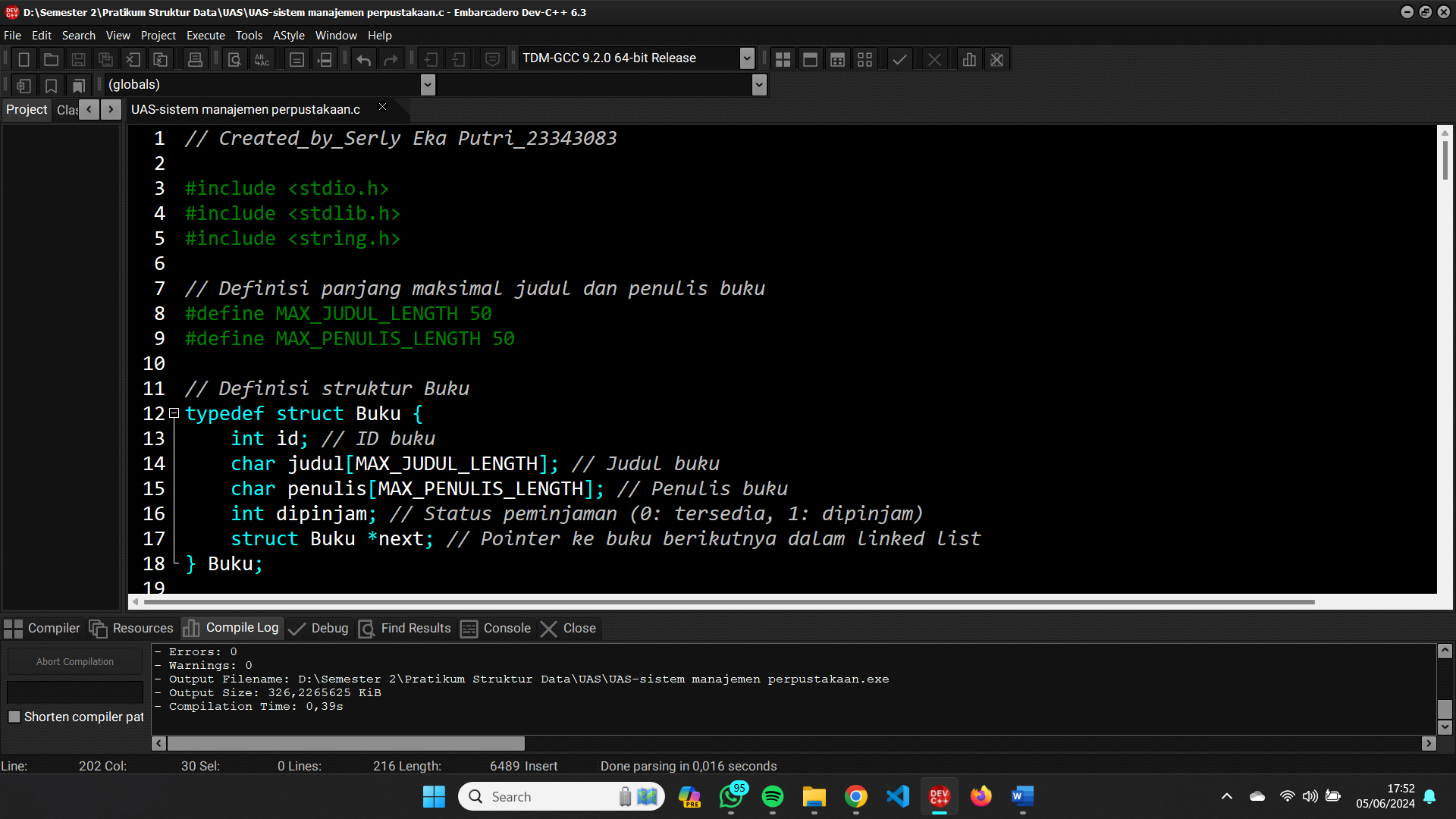




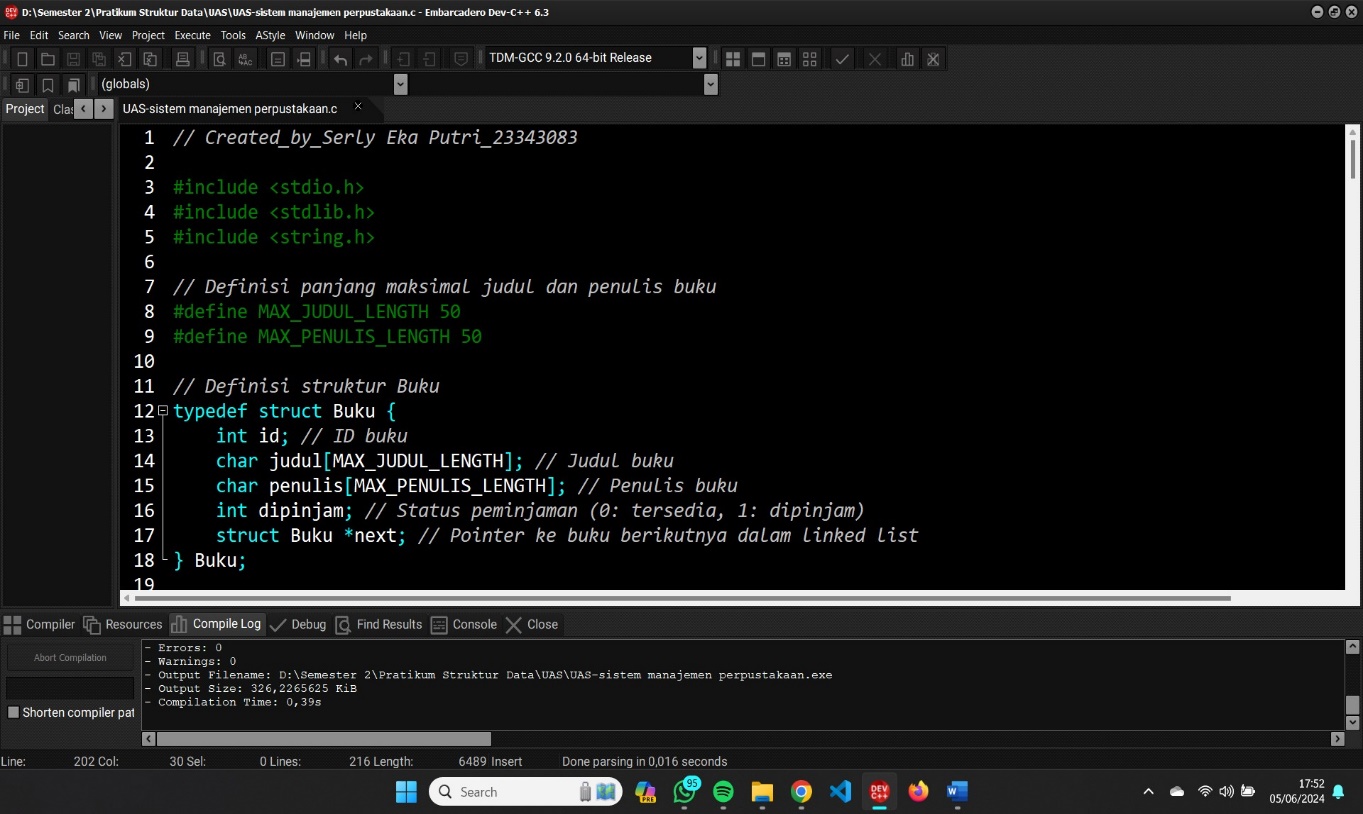
1. Penjelasan Program



1. #include <stdio.h> : Mengimpor Pustaka standar input-output untuk fungsi seperti ‘printf’ dan ‘scanf’.
2. #include <stdlib.h> : Mengimpor Pustaka standar untuk fungsi seperti ‘malloc’ dan ‘free’.
3. #include <string.h> : Mengimpor Pustaka string untuk fungsi seperti ‘strcpy’ dan ‘strstr’



1. #define MAX\_JUDUL\_LENGTH 50 : Mendefinisikan Panjang maksimun untuk judul buku sebagai 50 karakter.
2. #define MAX\_PENULIS\_LENGTH 50 :Mendefinisikan Panjang maksimun untuk penulis buku sebagai 50 karakter.



1. Typedef struct buku {…} buku; : Mendefinisikan struktur ‘Buku’ dengan beberapa atribut: ‘id’, ‘judul’, ‘penulis’, ‘dipinjam’, dan pointer ‘next’ yang menunjuk ke buku berikutnya dalam linked list.
2. ‘Buku \*head = NULL; ‘ : M<endefinisikan pointer ‘head’ yang menunjukkan ke buku pertama dalam linked list.
3. ‘Buku \*head = NULL; ‘ : M<endefinisikan pointer ‘head’ yang menunjukkan ke buku pertama dalam linked list.
4. ‘ int jumlahBuku = 0;: Mendefinisikan variabel jumlahBuku untuk menyimpan jumlah total buku dalam perpustakaan.
5. void tambahBuku() { /\* ... \*/ }: Mendefinisikan fungsi tambahBuku() untuk menambahkan buku baru ke dalam perpustakaan.
6. void pinjamBuku() { /\* ... \*/ }: Mendefinisikan fungsi pinjamBuku() untuk meminjam buku dari perpustakaan.
7. void kembalikanBuku() { /\* ... \*/ }: Mendefinisikan fungsi kembalikanBuku() untuk mengembalikan buku yang sudah dipinjam.
8. void cariBuku() { /\* ... \*/ }: Mendefinisikan fungsi cariBuku() untuk mencari buku berdasarkan judul atau penulis.
9. void daftarBuku() { /\* ... \*/ }: Mendefinisikan fungsi daftarBuku() untuk menampilkan daftar semua buku dalam perpustakaan.
10. void tukarBuku(Buku a, Buku \*b) { / ... \*/ }: Mendefinisikan fungsi tukarBuku() untuk menukar informasi antara dua buku.
11. void sortBuku() { /\* ... \*/ }: Mendefinisikan fungsi sortBuku() untuk mengurutkan buku dalam perpustakaan berdasarkan judul.
12. int main() { /\* ... \*/ }: Mendefinisikan fungsi utama main() yang akan menjalankan program.
13. int pilihan;: Mendeklarasikan variabel pilihan untuk menyimpan pilihan menu dari pengguna.
14. while (1) { /\* ... \*/ }: Memulai loop tak terbatas untuk menampilkan menu dan memproses pilihan pengguna.
15. printf("\nSistem Manajemen Perpustakaan\n");: Menampilkan judul program.
16. scanf("%d", &pilihan);: Meminta pengguna untuk memasukkan pilihan menu.
17. switch (pilihan) { /\* ... \*/ }: Memulai struktur switch-case untuk memproses pilihan pengguna.
18. case 1: tambahBuku(); break;: Memanggil fungsi tambahBuku() jika pengguna memilih opsi untuk menambahkan buku.
19. case 2: pinjamBuku(); break;: Memanggil fungsi pinjamBuku() jika pengguna memilih opsi untuk meminjam buku.
20. case 3: kembalikanBuku(); break;: Memanggil fungsi kembalikanBuku() jika pengguna memilih opsi untuk mengembalikan buku.
21. case 4: cariBuku(); break;: Memanggil fungsi cariBuku() jika pengguna memilih opsi untuk mencari buku.
22. case 5: daftarBuku(); break;: Memanggil fungsi daftarBuku() jika pengguna memilih opsi untuk menampilkan daftar semua buku.
23. case 6: sortBuku(); break;: Memanggil fungsi sortBuku() jika pengguna memilih opsi untuk mengurutkan buku berdasarkan judul.
24. Kesimpulan

System manajemen perpustakaan yang diimplementasikan dalam program ini menyediakan antarmuka sederhana untuk pengelolaan buku dalam perpustakaan.

Berikut ini adalah beberapa Kesimpulan dari system ini :

1. Kemudahan pengguna :

Sistem menyediakan menu sederhana yang memungkinkan pengguna dengan mudah melakukan operasi seperti menambahkan buku baru, meminjam buku, mengembalikan buku, mencari buku berdasarkan judul atau penulis, menampilkan daftar semua buku, dan mengurutkan buku berdasarkan judul.

1. Manajemen buku :

Setiap buku direpresentasikan sebagai sebuah entitas dengan atribut seperti judul, penulis, ID, dan status peminjaman. Hal ini memungkinkan sistem untuk melakukan operasi-operasi yang diperlukan seperti meminjam dan mengembalikan buku dengan mudah.

1. Pencarian Efesien :

Sistem memungkinkan pengguna untuk mencari buku berdasarkan judul atau penulis dengan memasukkan kata kunci pencarian. Ini memberikan kemudahan bagi pengguna untuk menemukan buku yang mereka cari.

1. Tampilan Informasi :

Fungsi daftarBuku() memberikan daftar semua buku dalam perpustakaan beserta informasi penting seperti judul, penulis, ID, dan status peminjaman. Ini memungkinkan pengguna untuk melihat informasi buku dengan mudah.

1. Pengurutan Buku :

Sistem dapat mengurutkan buku dalam perpustakaan berdasarkan judul secara alfabetis menggunakan fungsi sortBuku(). Ini membantu pengguna dalam menavigasi dan menemukan buku dengan lebih efisien.

1. Kontrol Kesalahan :

Sistem memiliki mekanisme untuk menangani kesalahan seperti gagal alokasi memori saat menambahkan buku baru atau meminjam buku yang tidak ada dalam perpustakaan.

Dengan demikian, Kesimpulan dari system manajemen perpustakaan ini adalah menyediakan antarmuka yang mudah digunakan, efisien dalam pencarian dan manajemen buku, serta memberikan fumgsionalitas dasar yang diperlukan dalam sebuah perpustakaan. System manajemen perpustakaan ini menyediakan Solusi yang efesien, dapat diandalkan, dan mudahdigunakan untuk mengelola perpustakaan dengan baik. Sytem ini dapat membantu meningkatkan efesiensi dan keteratruran dalam pengelolaan perpustakaan.