**PENJELASAN UTS NO. 5**

Salsabila Nur Aqmarina Aulia

2209076041

UTS Struktur Data

Pada Program manajemen perpustakaan sederhana ini membutuhkan penggunaan berbagai struktur data (*array, pointer, stack,* dan *queue*) untuk mengelola data buku, peminjaman, serta riwayat peminjaman. Berikut adalah penjelasan untuk masing-masing fitur yang diminta dalam soal:

**1. Menggunakan Struct `Buku` dengan Anggota `ISBN`, `judul`, `pengarang`, dan `tahunTerbit`**

Struct `Buku` digunakan untuk menyimpan informasi setiap buku, yaitu:

- `ISBN`: Nomor identifikasi unik buku.

- `judul`: Judul buku.

- `pengarang`: Nama pengarang buku.

- `tahunTerbit`: Tahun terbit buku.

Contoh deklarasi struct `Buku`:

struct Buku {

std::string ISBN;

std::string judul;

std::string pengarang;

int tahunTerbit;

};

```

**2. Menyimpan Data Buku Menggunakan Array of Pointers**

Kita dapat menggunakan array pointer untuk menyimpan alamat dari struct `Buku`. Ini memungkinkan pengelolaan data buku secara dinamis. Misalnya, kita bisa menyimpan hingga 100 buku dengan `Buku\* koleksiBuku[100]`, di mana setiap elemen adalah pointer ke struct `Buku`.

Pada C++ :

Buku\* koleksiBuku[100]; // Array of pointers untuk data buku

int jumlahBuku = 0; // Menyimpan jumlah buku saat ini

```

**3. Mengimplementasikan Stack untuk Menyimpan Riwayat Peminjaman Buku**

Untuk mencatat riwayat peminjaman buku, kita menggunakan stack. Setiap kali buku dipinjam, data peminjaman (misalnya ISBN buku dan waktu pinjam) akan disimpan di stack. Ini memungkinkan pelacakan riwayat peminjaman terakhir hingga peminjaman pertama.

Pada C++ :

#include <stack>

struct RiwayatPeminjaman {

std::string ISBN;

std::string waktuPeminjaman;

};

std::stack<RiwayatPeminjaman> riwayatPeminjaman; // Stack untuk riwayat peminjaman

```

**4. Menggunakan Queue untuk Mengelola Antrian Peminjaman Buku**

Queue digunakan untuk mengelola antrian peminjaman buku. Setiap kali ada permintaan peminjaman, data pelanggan akan dimasukkan ke antrian. Pelanggan yang berada di depan antrian akan dilayani lebih dulu.

Pada C++ :

#include <queue>

struct AntrianPeminjaman {

std::string ISBN;

std::string namaPeminjam;

};

std::queue<AntrianPeminjaman> antrianPeminjaman; // Queue untuk antrian peminjaman buku

```

**5. Membuat Fungsi untuk Menambah, Mencari, dan Menampilkan Data Buku**

a) Fungsi Menambah Buku : Menyimpan data buku baru ke dalam array pointer `koleksiBuku`.

Pada C++ :

void tambahBuku(const std::string& ISBN, const std::string& judul, const std::string& pengarang, int tahun) {

if (jumlahBuku < 100) {

koleksiBuku[jumlahBuku] = new Buku{ISBN, judul, pengarang, tahun};

jumlahBuku++;

std::cout << "Buku berhasil ditambahkan!\n";

} else {

std::cout << "Koleksi buku penuh!\n";

}

}

```

b) Fungsi Mencari Buku : Mencari buku berdasarkan ISBN dan menampilkan data buku jika ditemukan.

Pada C++ :

Buku\* cariBuku(const std::string& ISBN) {

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

if (koleksiBuku[i]->ISBN == ISBN) {

return koleksiBuku[i];

}

}

return nullptr;

}

```

c) Fungsi Menampilkan Buku : Menampilkan semua data buku dalam koleksi.

Pada C++ :

void tampilkanKoleksiBuku() {

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

std::cout << "ISBN: " << koleksiBuku[i]->ISBN << ", Judul: " << koleksiBuku[i]->judul

<< ", Pengarang: " << koleksiBuku[i]->pengarang << ", Tahun Terbit: " << koleksiBuku[i]->tahunTerbit << "\n";

}

}

```

**6. Mengimplementasikan Fungsi Peminjaman dan Pengembalian Buku**

1. Fungsi Peminjaman Buku : Memasukkan pelanggan ke dalam `antrianPeminjaman`. Saat pelanggan mengambil buku, data riwayat peminjaman disimpan ke `riwayatPeminjaman`.

Pada C++ :

void pinjamBuku(const std::string& ISBN, const std::string& namaPeminjam) {

Buku\* buku = cariBuku(ISBN);

if (buku) {

antrianPeminjaman.push({ISBN, namaPeminjam});

std::cout << "Permintaan peminjaman buku \"" << buku->judul << "\" telah ditambahkan ke antrian.\n";

} else {

std::cout << "Buku tidak ditemukan.\n";

}

}

void prosesPeminjaman() {

if (!antrianPeminjaman.empty()) {

AntrianPeminjaman peminjaman = antrianPeminjaman.front();

antrianPeminjaman.pop();

// Tambahkan riwayat peminjaman ke stack

riwayatPeminjaman.push({peminjaman.ISBN, "Tanggal/Waktu"});

std::cout << "Buku dengan ISBN " << peminjaman.ISBN << " dipinjam oleh " << peminjaman.namaPeminjam << "\n";

} else {

std::cout << "Tidak ada antrian peminjaman.\n";

}

}

```

b) Fungsi Pengembalian Buku : Menghapus riwayat peminjaman dari stack (riwayat terbaru dikeluarkan terlebih dahulu sesuai sifat stack).

Pada C++:

void kembalikanBuku() {

if (!riwayatPeminjaman.empty()) {

RiwayatPeminjaman terakhir = riwayatPeminjaman.top();

riwayatPeminjaman.pop();

std::cout << "Buku dengan ISBN " << terakhir.ISBN << " telah dikembalikan.\n";

} else {

std::cout << "Tidak ada riwayat peminjaman untuk dikembalikan.\n";

}

}

```

**7. Program Utama**

Program utama mengelola alur input pengguna, seperti menambah buku, melakukan peminjaman, dan menampilkan koleksi buku.

Pada C++ :

int main() {

tambahBuku("12345", "Pemrograman C++", "John Doe", 2020);

tambahBuku("67890", "Pemrograman Python", "Jane Smith", 2018);

int pilihan;

do {

std::cout << "\nMenu Perpustakaan:\n";

std::cout << "1. Tambah Buku\n";

std::cout << "2. Cari Buku\n";

std::cout << "3. Tampilkan Koleksi Buku\n";

std::cout << "4. Pinjam Buku\n";

std::cout << "5. Proses Antrian Peminjaman\n";

std::cout << "6. Kembalikan Buku\n";

std::cout << "0. Keluar\n";

std::cout << "Pilihan: ";

std::cin >> pilihan;

if (pilihan == 1) {

std::string ISBN, judul, pengarang;

int tahun;

std::cout << "Masukkan ISBN: ";

std::cin >> ISBN;

std::cout << "Masukkan Judul: ";

std::cin.ignore();

std::getline(std::cin, judul);

std::cout << "Masukkan Pengarang: ";

std::getline(std::cin, pengarang);

std::cout << "Masukkan Tahun Terbit: ";

std::cin >> tahun;

tambahBuku(ISBN, judul, pengarang, tahun);

} else if (pilihan == 4) {

std::string ISBN, namaPeminjam;

std::cout << "Masukkan ISBN: ";

std::cin >> ISBN;

std::cout << "Masukkan Nama Peminjam: ";

std::cin.ignore();

std::getline(std::cin, namaPeminjam);

pinjamBuku(ISBN, namaPeminjam);

}

// Tambahkan logika lainnya sesuai pilihan

} while (pilihan != 0);

return 0;

}

```

Jadi dengan mengimplementasikan nya program perpustakaan dapat menambah buku, mencari buku, memproses peminjaman dan pengembalian, serta mencatat riwayat peminjaman menggunakan stack dan antrian peminjaman dengan queue.