

Nama : Fikyarlos Mark
NIM : 2209076053
Mata Kuliah : Struktur Data

Laporan Penjelasan Program

1. Array dan Pointer

Program soal no. 1 merupakan aplikasi mengelolah data mahasiswa menggunakan bahasa pemrograman C++. Aplikasi ini menyediakan fitur untuk menambah, menghapus, mengurutkan, dan menampilkan data mahasiswa yang terdiri dari NIM, nama dan IPK. Program ini memungkinkan pengguna untuk mengelola data secara efisien melalui berbagai fungsi yang telah dirancang.

Program ini menggunakan struktur Mahasiswa untuk menyimpan informasi setiap mahasiswa. Struktur ini memiliki tiga atribut, yaitu NIM, nama, dan IPK, yang digunakan untuk mengidentifikasi dan menyimpan data dasar dari masing-masing mahasiswa. Dengan menggunakan struct ini, program dapat menyimpan data secara terstruktur dan mudah diakses.

Fungsi `tambahMahasiswa` digunakan untuk menambahkan data mahasiswa baru ke dalam array `arr`. Fungsi ini meminta input dari pengguna berupa NIM, nama, dan IPK. Jika jumlah mahasiswa belum mencapai batas maksimum (10 mahasiswa), data akan disimpan, dan variabel jumlah akan bertambah. Jika data sudah penuh, pengguna akan menerima pesan bahwa tidak dapat menambah data baru.

Fungsi `hapusMahasiswa` berfungsi untuk menghapus data mahasiswa berdasarkan NIM yang dimasukkan oleh pengguna. Fungsi ini akan mencari mahasiswa dengan NIM yang sesuai, menghapus data tersebut dari memori, dan menyesuaikan posisi array sehingga tidak ada data yang kosong di tengah-tengah. Jika mahasiswa dengan NIM tersebut tidak ditemukan, program akan menampilkan pesan bahwa NIM tidak ada dalam data.

Fungsi `tampilkanMahasiswa` digunakan untuk menampilkan data mahasiswa,. Fungsi ini akan mencetak semua data yang tersimpan dalam bentuk tabel dengan kolom yang terdiri dari NIM, Nama, dan IPK. Jika tidak ada data mahasiswa yang tersimpan, program akan memberikan informasi bahwa data tidak ada.

Selain itu, program juga menyediakan fungsi `urutkanMahasiswa` yang digunakan untuk mengurutkan data mahasiswa berdasarkan IPK dari yang tertinggi ke yang terendah menggunakan algoritma *Bubble Sort*. Setelah data diurutkan, pengguna dapat memilih menu untuk melihat hasil urutan tersebut.

Fungsi utama (`main`) mengontrol alur aplikasi dan menampilkan menu pilihan kepada pengguna. Program dimulai dengan meminta jumlah mahasiswa yang akan diinput, kemudian menawarkan menu dengan beberapa pilihan, yaitu:

1. Tambah Data Mahasiswa
2. Hapus Data Mahasiswa
3. Urutkan Data Mahasiswa Berdasarkan IPK
4. Tampilkan Data Mahasiswa
5. Keluar

Pengguna dapat memilih salah satu dari opsi ini untuk menjalankan fungsi yang diinginkan. Program akan terus beroperasi dan menampilkan menu sampai pengguna memilih opsi untuk keluar, yaitu pilihan ke-5. Setelah program selesai dijalankan, memori yang dialokasikan untuk objek `Mahasiswa` akan dibebaskan melalui perulangan `delete` pada setiap objek dalam array. Hal ini bertujuan untuk mencegah kebocoran memori dan memastikan program berjalan secara efisien.

2. Struct dan File Handling

Program soal no. 2 adalah aplikasi manajemen inventaris peralatan menggunakan bahasa C++. Program ini menyimpan data peralatan dalam bentuk vector dan memberikan fitur untuk menambah, mengubah, menghapus, menampilkan, dan menyimpan data ke dalam file. Fitur-fitur ini membantu dalam mengelola informasi inventaris secara efisien dan terstruktur. Program dimulai dengan mendefinisikan struct Peralatan, yang memiliki empat atribut: kode (string), nama (string), jumlah (integer), dan kondisi (string). Struktur ini digunakan untuk menyimpan informasi dasar tentang setiap peralatan.

Penjelasan Fitur Program:

1. Tambah Peralatan

Fungsi `tambahPeralatan` digunakan untuk menambahkan data peralatan baru ke dalam inventaris. Pengguna diminta untuk memasukkan kode, nama, jumlah, dan kondisi peralatan. Data yang diinput akan disimpan dalam objek Peralatan, yang kemudian ditambahkan ke dalam vector inventaris. Fitur ini memungkinkan pengguna untuk menambah data dengan mudah.

2. Ubah Peralatan

Fungsi `ubahPeralatan` memungkinkan pengguna untuk mengubah informasi peralatan yang sudah ada berdasarkan kode yang diberikan. Setelah pengguna memasukkan kode peralatan yang ingin diubah, program mencari data yang sesuai. Jika ditemukan, program menampilkan data saat ini dan meminta pengguna untuk memasukkan data baru. Setelah proses pengubahan selesai, data di dalam vector akan diperbarui. Jika tidak ditemukan, pesan kesalahan akan ditampilkan.

3. Hapus Peralatan

Fungsi `hapusPeralatan` digunakan untuk menghapus data peralatan dari inventaris berdasarkan kode. Program mencari data yang sesuai dengan kode dan menghapusnya menggunakan algoritma `remove_if`. Jika peralatan ditemukan dan dihapus, program akan memberi pesan bahwa data berhasil dihapus. Jika tidak, program akan menampilkan pesan bahwa data tidak ditemukan.

4. Simpan Data ke File

Fungsi `simpanKeFile` menyimpan data inventaris ke dalam file teks dengan format terstruktur menggunakan karakter pemisah '|'. Setiap baris dalam file mewakili satu data peralatan. Ini memudahkan pengguna untuk menyimpan data secara permanen dan memuatnya kembali ketika program dijalankan lagi. Jika file gagal dibuka, program akan menampilkan pesan kesalahan.

5. Baca Data dari File

Fungsi `bacaDariFile` memuat data peralatan dari file teks saat program pertama kali dijalankan. Program membaca isi file, memisahkan data berdasarkan karakter pemisah, dan menyimpannya dalam vector inventaris. Ini memastikan bahwa data inventaris terakhir yang disimpan dapat diakses kembali ketika program dibuka.

6. Tampilkan Laporan Inventaris Peralatan

Fungsi `tampilkanLaporan` menampilkan semua data peralatan dalam bentuk tabel dengan kolom yang terdiri dari kode, nama, jumlah, dan kondisi. Sebelum menampilkan, data diurutkan berdasarkan kode untuk memudahkan pencarian dan pemahaman. Jika tidak ada data dalam inventaris, program akan memberikan informasi bahwa tidak ada data yang tersedia.

Alur Program:

Program diawali dengan memuat data dari file `inventaris.txt`. Setelah itu, program menampilkan menu dengan pilihan berikut:

1. Tambah Peralatan
2. Ubah Peralatan
3. Hapus Peralatan
4. Tampilkan Laporan
5. Simpan ke File
6. Keluar

Pengguna dapat memilih opsi yang sesuai untuk melakukan operasi tertentu. Program akan terus beroperasi dan menampilkan menu sampai pengguna memilih opsi "Keluar". Sebelum program selesai, pengguna juga dapat menyimpan perubahan data ke file menggunakan opsi "Simpan ke File".

3. Stuck

Program soal no. 3 adalah aplikasi untuk mengevaluasi ekspresi aritmatika dalam bentuk postfix (notasi Polandia terbalik) menggunakan **stack**. Program menerima input berupa ekspresi postfix, kemudian mengevaluasi hasilnya menggunakan operator dan operand yang ditangani oleh stack.

Penjelasan Fitur Program:

1) Class Stack Menggunakan STL stack

Class Stack adalah implementasi pembungkus dari stack standar (STL) di C++. Stack adalah struktur data LIFO (Last In, First Out) yang digunakan untuk menyimpan operand dan hasil operasi selama evaluasi ekspresi postfix.

- **Method push(double value):** Menambahkan nilai ke dalam stack.
- **Method pop():** Mengambil nilai dari stack, sekaligus menghapusnya. Jika stack kosong, program menampilkan pesan kesalahan.
- **Method isEmpty():** Mengecek apakah stack kosong.
- **Method top():** Mengembalikan nilai di bagian atas stack tanpa menghapusnya.

2) Fungsi isOperator

Fungsi ini digunakan untuk mengecek apakah karakter yang diberikan adalah operator aritmatika, yaitu: +, -, *, /, atau ^ (pangkat). Fungsi ini memudahkan dalam membedakan antara operator dan operand saat evaluasi postfix.

3) Fungsi evaluasiPostfix

Fungsi ini adalah inti dari program, yang bertugas mengevaluasi ekspresi postfix. Beberapa langkah yang dilakukan fungsi ini adalah:

- Memecah ekspresi postfix ke dalam token-token menggunakan stringstream.
- Setiap token dicek apakah berupa operand (angka) atau operator:
 - Jika token adalah angka, ia dimasukkan ke dalam stack.
 - Jika token adalah operator, dua angka terakhir yang ada di dalam stack diambil, lalu dilakukan operasi sesuai operator tersebut (misalnya penjumlahan, pengurangan, dll.). Hasil operasi kemudian dimasukkan kembali ke dalam stack.
- Fungsi ini juga memberikan log langkah-langkah evaluasi secara rinci ke layar.

- Hasil akhir evaluasi adalah nilai yang tersisa di dalam stack setelah semua operasi dilakukan.

Proses Evaluasi Postfix:

1. Input Ekspresi Postfix

Pengguna diminta untuk memasukkan ekspresi postfix dengan angka dan operator yang dipisahkan oleh spasi. Misalnya, ekspresi $3\ 4\ +\ 2\ *\ 7\ /\$ akan dihitung sebagai:

- $(3 + 4) * 2 / 7$

2. Pengecekan dan Proses Operand atau Operator

- Jika token yang dibaca adalah angka, maka akan di-*push* ke stack.
- Jika token adalah operator, dua operand terakhir akan di-*pop* dari stack dan operasi sesuai token (operator) akan dilakukan. Hasil operasi akan dimasukkan kembali ke stack.

Misalnya:

- Pada ekspresi $3\ 4\ +$, angka 3 dan 4 di-*push* ke stack. Operator + kemudian mengambil kedua angka tersebut dan menjumlahkannya ($3 + 4 = 7$), dan hasilnya 7 di-*push* kembali ke stack.

3. Pengelolaan Operator

- Untuk operator penjumlahan (+), pengurangan (-), perkalian (*), dan pembagian (/), fungsi melakukan operasi aritmatika standar.
- Untuk operator pangkat (^), program menggunakan fungsi `pow()` dari pustaka matematika C++ (`cmath`).

4. Penanganan Pembagian dengan Nol

Jika operasi pembagian melibatkan pembagian dengan nol, program akan menampilkan pesan kesalahan "Error: Pembagian dengan nol" dan mengembalikan nilai 0.

5. Hasil Akhir Evaluasi

Setelah semua token dalam ekspresi postfix diproses, nilai akhir yang ada di stack adalah hasil evaluasi ekspresi. Fungsi `pop()` akan mengembalikan nilai ini dan menampilkan hasilnya.

Alur Program:

- Program meminta pengguna memasukkan ekspresi postfix.
- Fungsi `evaluasiPostfix` memproses ekspresi tersebut, memecahnya menjadi token, mengevaluasi tiap token (operand atau operator), dan melakukan operasi aritmatika yang sesuai.
- Hasil akhir evaluasi ditampilkan ke layar.

4. Queue

Program soal no. 4 merupakan simulasi sistem antrian yang menggunakan struktur data **queue** (antrian) untuk memproses layanan di tiga loket secara bersamaan. Program ini menerima input berupa jumlah pelanggan serta waktu layanan untuk masing-masing pelanggan, kemudian memproses antrian tersebut di tiga loket yang tersedia.

Struktur data yang digunakan adalah **struct Pelanggan** untuk menyimpan data pelanggan dengan atribut seperti nomorAntrian (string) yang berfungsi sebagai identitas unik pelanggan, dan waktuLayanan (int) yang mencatat durasi layanan yang dibutuhkan dalam menit. Class Queue mengelola antrian pelanggan dengan menyediakan fungsi seperti enqueue untuk menambahkan pelanggan ke antrian, dequeue untuk mengambil pelanggan dari antrian, serta fungsi pendukung seperti isEmpty untuk mengecek apakah antrian kosong, dan size untuk mengetahui jumlah pelanggan dalam antrian.

Fungsi utama dalam program ini adalah `simulasiAntrian`, yang menjalankan proses layanan di tiga loket. Setiap loket memiliki variabel `waktuLoket` yang melacak sisa waktu layanan. Apabila suatu loket kosong (tidak ada pelanggan yang sedang dilayani), pelanggan dari antrian akan diproses di loket tersebut. Waktu layanan pada setiap loket dikurangi setiap iterasi, dan saat waktu mencapai nol, loket menjadi kosong dan siap melayani pelanggan berikutnya. Simulasi ini terus berjalan hingga semua pelanggan selesai dilayani, dan tidak ada pelanggan yang tersisa dalam antrian atau sedang dilayani. Setelah selesai, program akan menampilkan statistik berupa jumlah total pelanggan yang terlayani, rata-rata waktu tunggu pelanggan, dan jumlah sisa pelanggan yang belum dilayani (jika ada).

Pada awal program, pengguna akan diminta memasukkan jumlah pelanggan, serta nomor antrian dan waktu layanan masing-masing pelanggan. Setiap pelanggan kemudian ditambahkan ke antrian melalui fungsi enqueue. Saat simulasi dijalankan, pelanggan akan diproses di loket yang tersedia dan laporan mengenai proses layanan ditampilkan secara rinci. Setelah simulasi selesai, statistik akan diberikan untuk menunjukkan efisiensi layanan, termasuk total pelanggan terlayani, rata-rata waktu tunggu, dan sisa antrian.

5. Implementasi Gabungan

Program soal no. 5 adalah simulasi sistem manajemen perpustakaan yang memungkinkan pengguna mengelola buku-buku perpustakaan, termasuk menambah, mencari, meminjam, mengembalikan, serta menampilkan daftar buku yang tersedia. Program menggunakan beberapa struktur data penting seperti **array of pointers**, **queue**, dan **stack** untuk menyusun data buku, antrian peminjaman, serta riwayat peminjaman. Data buku disimpan dalam struct Buku, yang mencakup ISBN, judul, pengarang, dan tahun terbit. Koleksi buku dikelola menggunakan array of pointers bernama daftarBuku, dengan kapasitas maksimum 100 buku. Untuk memudahkan proses peminjaman, program menggunakan struktur **queue** yang menempatkan setiap peminjam dalam antrian berdasarkan urutan kedatangan, sementara **stack** digunakan untuk mencatat riwayat peminjaman buku, dengan buku yang baru saja dipinjam berada di urutan paling atas.

Fungsi Utama dalam Program:

1. **tambahBuku:**

Menambahkan buku baru ke dalam koleksi perpustakaan. Buku baru disimpan dalam array of pointers daftarBuku.

2. **cariBuku:**

Mencari buku berdasarkan ISBN yang diberikan oleh pengguna. Jika ditemukan, program akan menampilkan informasi buku tersebut.

3. **tampilkanSemuaBuku:**

Menampilkan seluruh koleksi buku yang ada di perpustakaan dengan informasi detail seperti ISBN, judul, pengarang, dan tahun terbit.

4. **pinjamBuku:**

Menambahkan buku ke dalam antrian peminjaman dan mencatatnya dalam riwayat peminjaman. Jika buku dengan ISBN yang diberikan ditemukan di koleksi perpustakaan, maka buku tersebut akan masuk ke dalam antrian.

5. **kembalikanBuku:**

Menghapus buku dari antrian peminjaman, menandakan bahwa buku tersebut telah dikembalikan.

6. **tampilkanRiwayat:**

Menampilkan riwayat peminjaman buku dalam urutan terakhir dipinjam ke yang paling awal.

7. **tampilkanAntrian:**

Menampilkan daftar buku yang sedang menunggu untuk dipinjam di antrian.

Mekanisme Program:

1. Program dimulai dengan menambahkan beberapa buku contoh ke dalam perpustakaan.
2. Pengguna akan diberi pilihan menu untuk melakukan berbagai operasi seperti menambah buku baru, mencari buku berdasarkan ISBN, meminjam dan mengembalikan buku, serta melihat daftar antrian dan riwayat peminjaman.
3. **Saat meminjam buku:**
 - Program akan mengecek apakah buku dengan ISBN yang diminta tersedia di koleksi.
 - Jika ditemukan, buku akan ditambahkan ke antrian peminjaman dan riwayat peminjaman.
4. **Saat mengembalikan buku:**

Buku yang berada di urutan paling depan antrian akan dihapus, menandakan buku tersebut telah dikembalikan.
5. **Menampilkan riwayat dan antrian:**

Pengguna dapat melihat riwayat buku yang pernah dipinjam dan juga melihat daftar antrian buku yang sedang menunggu untuk dipinjam.