

PENJELASAN LAPORAN UTS STRUKTUR DATA

NAMA : ZHAFIF ANSHAR ZUHAYR

NIM 2209076001

1. soal pada nomer 1

Pada Coding nomer 1 merupakan program sederhana dalam bahasa pemrograman C++ yang berfungsi untuk ****mengelola data mahasiswa****. Program ini mencakup beberapa fitur seperti menambahkan data mahasiswa baru, menghapus data mahasiswa berdasarkan NIM, menampilkan semua data mahasiswa, serta mengurutkan data mahasiswa berdasarkan IPK secara menurun (dari nilai tertinggi ke terendah). Berikut ini adalah penjelasan mengenai beberapa istilah dan fungsi utama dalam coding tersebut:

1. Struct ``Mahasiswa: struct`` (struktur) digunakan untuk membuat tipe data yang terdiri dari beberapa variabel dengan tipe yang berbeda. Dalam hal ini, ``struct Mahasiswa`` digunakan untuk menyimpan data mahasiswa, yang mencakup ****NIM****, ****nama****, dan ****IPK****.

2. Array Pointer ``Mahasiswa`` mahasiswa adalah array yang berisi pointer ke objek ``Mahasiswa``. Program ini menggunakan array ini untuk menyimpan alamat dari setiap objek ``Mahasiswa`` yang dibuat secara dinamis dengan ``new``. Ukuran maksimum array ini adalah ``MAX`` (10), yang berarti hanya dapat menyimpan maksimal 10 mahasiswa.

3. Fungsi ``tambahMahasiswa`` digunakan untuk menambah data mahasiswa baru ke dalam array. Jika jumlah mahasiswa belum mencapai maksimum, fungsi akan meminta pengguna untuk memasukkan NIM, nama, dan IPK. Data mahasiswa ini kemudian ditambahkan ke array, dan variabel ``jumlah`` (jumlah mahasiswa saat ini) ditingkatkan.

4. Fungsi ``hapusMahasiswa`` berfungsi untuk menghapus data mahasiswa berdasarkan NIM. Fungsi ini mencari NIM yang diberikan pengguna dalam array. Jika NIM ditemukan, mahasiswa tersebut akan dihapus dengan ``delete``, lalu data mahasiswa lainnya akan digeser untuk menutup celah yang muncul akibat penghapusan.

5. Fungsi ``tampilkanMahasiswa`` menampilkan data semua mahasiswa yang tersimpan dalam array. Program ini akan mencetak setiap NIM, nama, dan IPK mahasiswa secara berurutan.

6. Fungsi `urutkanMahasiswa` mengurutkan data mahasiswa dalam array berdasarkan nilai IPK secara menurun (dari IPK tertinggi ke terendah) menggunakan metode Bubble Sort. Setelah selesai diurutkan, data mahasiswa akan dicetak dalam urutan baru.

7. Memori Dinamis Program ini menggunakan `new` untuk mengalokasikan memori bagi setiap mahasiswa yang ditambahkan, sehingga setiap objek `Mahasiswa` yang dibuat berada dalam memori heap. Di akhir program, objek-objek tersebut dihapus dengan `delete` untuk menghindari kebocoran memori.

Dengan Cara Pengoperasian

- Saat program berjalan, pengguna akan disajikan menu pilihan untuk melakukan berbagai operasi (seperti menambah, menghapus, menampilkan, atau mengurutkan mahasiswa).
- Pengguna memilih salah satu opsi, kemudian mengikuti instruksi untuk memasukkan atau melihat data mahasiswa.
- Jika pengguna memilih untuk keluar, program akan menghapus semua alokasi memori sebelum berhenti.

2. penjelasan soal pada nomer 2

Program di atas adalah aplikasi pengelolaan inventaris peralatan berbasis teks dalam C++ yang memungkinkan pengguna menambah, mengubah, menghapus, menampilkan, dan menyimpan data inventaris dalam file `inventaris.txt`. Program ini menggunakan struct `Peralatan` untuk mendefinisikan data peralatan dengan atribut `kode`, `nama`, `jumlah`, dan `kondisi`. Data inventaris disimpan secara dinamis dalam vector `inventaris` sehingga dapat menambah atau mengurangi data sesuai kebutuhan, sementara variabel `namaFile` menyimpan nama file untuk penyimpanan permanen. Saat program dimulai, fungsi `bacaDariFile` memuat data dari file, dan pengguna diberikan pilihan melalui menu untuk mengelola data peralatan: menambah dengan `tambahPeralatan`, memperbarui data melalui `ubahPeralatan`, atau menghapus data dengan `hapusPeralatan`. Fungsi `simpanKeFile` digunakan untuk menulis ulang semua data ke dalam file `inventaris.txt` agar perubahan tersimpan permanen, sedangkan `tampilkanInventaris` menampilkan data peralatan yang ada dalam urutan berdasarkan `kode`. Program ini menunjukkan konsep dasar pengelolaan data inventaris dengan vector, file handling, dan penggunaan fungsi modular dalam C++.

3. penjelasan soal pada nomer 3

program untuk mengevaluasi ekspresi postfix dalam bahasa C++, dengan penggunaan `stack` untuk memproses operasi aritmatika secara berurutan sesuai aturan postfix. Dalam ekspresi postfix, operator ditempatkan setelah operand (contoh: ``3 4 +`` untuk ``3 + 4``), sehingga evaluasinya dapat dilakukan tanpa perlu mempertimbangkan tanda kurung. Program ini meminta input berupa ekspresi postfix dari pengguna, kemudian menampilkan setiap langkah evaluasi serta hasil akhirnya.

Kelas `Stack` yang digunakan di sini menyediakan metode `push` untuk menambah nilai ke stack, `pop` untuk menghapus dan mengembalikan nilai paling atas dari stack, serta `isEmpty` untuk memeriksa apakah stack kosong. Dalam fungsi `evaluasiPostfix`, ekspresi postfix dipecah menjadi token-token yang dipisahkan oleh spasi, menggunakan `istringstream`. Jika token adalah angka, ia dikonversi ke `float` dan ditambahkan ke stack. Jika token adalah operator (misalnya `+`, `-`, `*`, `/`, atau `^` untuk perpangkatan), program mengambil dua nilai teratas dari stack (operand pertama dan kedua), lalu melakukan operasi yang sesuai, misalnya `a + b` untuk penjumlahan atau `pow(a, b)` untuk perpangkatan. Hasil operasi ini dikembalikan ke stack, yang memungkinkan evaluasi ekspresi berkelanjutan.

Selain itu, fungsi `tampilkanLangkah` berfungsi menampilkan proses evaluasi per langkah, memperlihatkan setiap operasi `push` dan hasil perhitungan. Fungsi ini memudahkan pengguna untuk memahami bagaimana program memanipulasi stack, dengan menampilkan setiap operasi aritmatika yang dilakukan (seperti `3 + 4 = 7`) sebelum hasilnya dikembalikan ke stack. Program ini menggabungkan konsep stack dan ekspresi postfix, di mana stack membantu menyimpan nilai operand secara sementara dan mengeksekusi operasi secara efisien, menyelesaikan ekspresi postfix tanpa memerlukan tanda kurung.

4. penjelasan soal pada nomer 4

program simulasi antrian layanan dengan beberapa loket, menggunakan struktur `queue` dalam C++ untuk mengatur urutan antrian pelanggan. Struktur `Pelanggan` didefinisikan dengan atribut `nomorAntrian` dan `waktuLayanan`, mewakili nomor antrian pelanggan dan estimasi waktu yang dibutuhkan untuk dilayani. Kelas `Queue` berisi metode `enqueue` untuk menambah pelanggan ke dalam antrian, `dequeue` untuk mengambil pelanggan dari antrian, serta `isEmpty` dan `size` untuk memeriksa apakah antrian kosong atau menghitung jumlah pelanggan dalam antrian. Di dalam fungsi `main`, pengguna diminta memasukkan jumlah pelanggan dan waktu layanan masing-masing pelanggan, yang kemudian disimpan dalam antrian `Queue`. Fungsi `simulasiAntrian` melakukan simulasi pelayanan di beberapa loket yang jumlahnya dapat diatur (dalam kasus ini, tiga loket). Setiap loket memiliki waktu selesai layanan yang dilacak dalam vector `loket`, yang diupdate setiap kali pelanggan dilayani. Program mencari loket yang siap paling cepat (waktu selesai terendah) untuk melayani pelanggan berikutnya, menghitung waktu

tunggu pelanggan berdasarkan selisih antara waktu selesai loket tersebut dengan jumlah pelanggan yang telah terlayani, dan menampilkan loket mana yang melayani pelanggan serta waktu tunggu masing-masing. Setelah semua pelanggan terlayani, program menampilkan statistik seperti rata-rata waktu tunggu, jumlah pelanggan yang terlayani, dan jumlah sisa pelanggan di antrian.

5. penjelasan coding pada latihan nomer 5

program manajemen perpustakaan sederhana yang menggunakan struktur data **stack** dan **queue** untuk mencatat riwayat peminjaman dan antrian peminjaman buku. Struktur ``Buku`` menyimpan informasi tentang buku seperti ``ISBN``, ``judul``, ``pengarang``, dan ``tahunTerbit``. Koleksi buku dikelola dalam bentuk **vector** untuk penyimpanan buku-buku yang tersedia. Program menyediakan beberapa fungsi: ``tambahBuku`` untuk menambah buku ke dalam koleksi perpustakaan, ``cariBuku`` untuk mencari buku berdasarkan ``ISBN``, ``tampilkanKoleksi`` untuk menampilkan semua buku yang tersedia dalam koleksi, ``pinjamBuku`` untuk memindahkan buku dari koleksi ke **queue** peminjaman, dan ``kembalikanBuku`` untuk memindahkan buku dari antrian peminjaman ke dalam **stack** riwayat peminjaman, menandakan bahwa buku telah dikembalikan. Di dalam fungsi ``main``, terdapat menu interaktif yang memungkinkan pengguna memilih tindakan seperti menambah buku, mencari buku, menampilkan koleksi, meminjam, dan mengembalikan buku. Program menggunakan **dynamic memory allocation** untuk setiap buku, dan membersihkan memori tersebut setelah program selesai untuk mencegah **memory leak**.