

Nama : Muhammad Arya Bagaskara

Nim : 2209076045

Matkul : Struktur Data

### 1. *Array dan Pointer*

Pada soal ini, saya selaku mahasiswa membuat program untuk mengelola data mahasiswa menggunakan *struct* dan *array of pointers*. Pertama, *struct Mahasiswa* dibuat untuk menyimpan data utama yaitu NIM, Nama, dan IPK. Data mahasiswa disimpan dalam *array pointer* untuk memungkinkan alokasi dinamis, yang dapat menambah efisiensi memori. Fungsi `'tambahMahasiswa'` menambahkan data baru, `'hapusMahasiswa'` menghapus data berdasarkan NIM, `'tampilkanMahasiswa'` menampilkan seluruh data, dan `'urutkanMahasiswa'` mengurutkan mahasiswa berdasarkan IPK menggunakan algoritma bubble sort. Dengan pendekatan ini, program dapat menyimpan hingga 10 data mahasiswa dan melakukan pengelolaan data dengan fungsi yang terstruktur.

### 2. *Struct dan File Handling*

Soal nomor 2 ini melibatkan manajemen inventaris peralatan dengan menggunakan *struct* dan *file handling*. *Struct Peralatan* dibuat untuk menyimpan data yang terdiri dari kode, nama, jumlah, dan kondisi peralatan. Fungsi `'tambahPeralatan'` menambahkan data baru ke file teks dengan menggunakan mode *append* untuk menjaga agar data sebelumnya tetap ada. Fungsi `'tampilkanPeralatan'` membaca dan menampilkan isi file. Program ini berguna untuk menyimpan data inventaris secara permanen ke dalam file teks, sehingga data tidak hilang meski program dihentikan. Ini berguna untuk pengelolaan inventaris laboratorium yang memerlukan catatan fisik yang aman dan mudah diakses.

### 3. *Stack*

Pada soal ini nomor 3 yakni membuat program kalkulator notasi postfix yang dibuat menggunakan struktur data *stack*. *Class Stack* didefinisikan dengan fungsi `'push'`, `'pop'`, `'top'`, dan `'isEmpty'` untuk pengelolaan data di dalam stack. Fungsi `'evaluatePostfix'` mengambil ekspresi dalam notasi postfix dan mengevaluasinya dengan membaca karakter satu per satu. Jika karakter adalah angka, maka dimasukkan ke stack; jika operator, dua elemen diambil dari stack dan dihitung sesuai operatornya. Stack memudahkan evaluasi postfix karena sifatnya LIFO, sehingga operator beroperasi langsung pada dua operand terakhir. Program ini dapat menangani operator dasar seperti penjumlahan, pengurangan, perkalian, pembagian, dan pangkat.

#### 4. *Queue*

Pada soal ini, simulasi antrian layanan pelanggan di bank diimplementasikan menggunakan struktur data *queue*. Class *Queue* berisi operasi `'enqueue'`, `'dequeue'`, `'isEmpty'`, dan `'front'`, sementara struct *Pelanggan* menyimpan data pelanggan berupa nomor antrian dan waktu layanan. Program menyimulasikan antrian dengan pelanggan yang datang dan mendapatkan layanan berdasarkan urutan antrian. *Queue* cocok untuk simulasi ini karena karakteristik FIFO-nya (*First In, First Out*), yang memungkinkan pelanggan pertama dalam antrian untuk dilayani terlebih dahulu. Dengan pendekatan ini, bank dapat melacak jumlah pelanggan dalam antrian dan menghitung statistik seperti rata-rata waktu tunggu serta jumlah pelanggan yang telah dilayani.

#### 5. Implementasi Gabungan

Soal terakhir yakni menggabungkan berbagai struktur data, yaitu *array*, *pointer*, *struct*, *stack*, dan *queue* dalam program manajemen perpustakaan. Program memiliki struct *Buku* untuk menyimpan informasi tentang buku, *stack* untuk mencatat riwayat peminjaman buku, dan *queue* untuk mengelola antrian peminjaman. Fungsi `'tambahBuku'` menambahkan data buku baru ke dalam perpustakaan, `'pinjamBuku'` menambahkan buku ke dalam antrian peminjaman, `'kembalikanBuku'` mengeluarkan buku dari antrian dan menyimpannya dalam *stack* sebagai riwayat peminjaman, serta `'tampilkanRiwayat'` menampilkan semua buku yang telah dipinjamkan. Implementasi ini menunjukkan bagaimana berbagai struktur data dapat digunakan bersama-sama untuk membangun sistem yang lebih kompleks dan lengkap dalam pengelolaan data perpustakaan.