**PENJELASAN UTS NO. 4**

Salsabila Nur Aqmarina Aulia

2209076041

UTS Struktur Data

Soal ini meminta kita membuat simulasi antrian layanan pelanggan di bank menggunakan struktur data queue (antrian) dalam bahasa pemrograman C++ atau yang sejenis. Berikut adalah penjelasan dari setiap langkah yang diperlukan:

**1. Membuat Class `Queue` dengan Operasi `enqueue`, `dequeue`, dan `isEmpty`**

Class `Queue` ini akan menjadi struktur data untuk menyimpan pelanggan yang sedang mengantri. Operasi dasar yang dibutuhkan:

- `enqueue`: Menambahkan pelanggan ke belakang antrian.

- `dequeue`: Mengeluarkan pelanggan dari depan antrian.

- `isEmpty`: Memeriksa apakah antrian kosong.

Berikut adalah implementasi sederhana class `Queue` menggunakan `std::queue`:

#include <iostream>

#include <queue>

template <typename T>

class Queue {

private:

std::queue<T> data;

public:

void enqueue(T value) {

data.push(value);

}

T dequeue() {

if (!isEmpty()) {

T value = data.front();

data.pop();

return value;

} else {

throw std::runtime\_error("Antrian kosong!");

}

}

bool isEmpty() const {

return data.empty();

}

};

```

**2. Membuat `struct Pelanggan` dengan Anggota `nomorAntrian` dan `waktuLayanan`**

`struct Pelanggan` digunakan untuk menyimpan informasi pelanggan yang meliputi:

- `nomorAntrian`: Nomor antrian pelanggan.

- `waktuLayanan`: Waktu yang dibutuhkan untuk melayani pelanggan, dalam satuan menit.

Pada C++ :

struct Pelanggan {

int nomorAntrian;

int waktuLayanan;

};

```

**3. Mensimulasikan Proses Antrian dengan 3 Loket Layanan**

Untuk mensimulasikan proses antrian dengan 3 loket layanan, kita bisa membuat *array* atau *vector* yang menyimpan waktu tersisa untuk setiap loket. Setiap loket akan melayani pelanggan secara berurutan dalam antrian, dan waktu layanan mereka berkurang setiap iterasi simulasi hingga layanan selesai.

Pada C++ :

void simulasiAntrian(Queue<Pelanggan> &antrian, int jumlahLoket) {

int waktuTotal = 0;

int pelangganTerlayani = 0;

std::vector<int> loket(jumlahLoket, 0); // Menyimpan waktu layanan setiap loket

while (!antrian.isEmpty() || std::any\_of(loket.begin(), loket.end(), [](int w) { return w > 0; })) {

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

if (loket[i] == 0 && !antrian.isEmpty()) { // Jika loket kosong dan ada antrian

Pelanggan pelanggan = antrian.dequeue();

loket[i] = pelanggan.waktuLayanan;

std::cout << "Pelanggan " << pelanggan.nomorAntrian << " dilayani di loket " << i + 1 << " selama " << pelanggan.waktuLayanan << " menit.\n";

pelangganTerlayani++;

waktuTotal += pelanggan.waktuLayanan;

}

if (loket[i] > 0) loket[i]--; // Mengurangi waktu layanan loket jika sedang melayani

}

}

}

```

**4. Menghitung dan Menampilkan Statistik**

Statistik yang harus dihitung adalah:

- Rata-rata waktu tunggu pelanggan : Dihitung dengan membagi total waktu layanan dengan jumlah pelanggan terlayani.

- Jumlah pelanggan terlayani : Menghitung berapa banyak pelanggan yang telah dilayani dalam simulasi.

- Sisa antrian : Jumlah pelanggan yang masih menunggu di antrian setelah simulasi selesai.

Contoh implementasi untuk menghitung dan menampilkan statistik:

void tampilkanStatistik(int totalWaktuLayanan, int pelangganTerlayani, int sisaAntrian) {

std::cout << "\n--- Statistik Antrian ---\n";

std::cout << "Jumlah pelanggan terlayani: " << pelangganTerlayani << "\n";

std::cout << "Rata-rata waktu tunggu: " << (pelangganTerlayani > 0 ? totalWaktuLayanan / pelangganTerlayani : 0) << " menit\n";

std::cout << "Sisa pelanggan di antrian: " << sisaAntrian << "\n";

}

```

**5. Program Utama**

Program utama akan meminta input untuk jumlah pelanggan, nomor antrian, dan waktu layanan masing-masing pelanggan. Kemudian, program akan menjalankan simulasi antrian dan menampilkan statistik setelah simulasi selesai.

Pada C++ :

int main() {

Queue<Pelanggan> antrian;

int jumlahPelanggan;

int jumlahLoket = 3; // Jumlah loket layanan

int totalWaktuLayanan = 0;

int pelangganTerlayani = 0;

std::cout << "Masukkan jumlah pelanggan: ";

std::cin >> jumlahPelanggan;

// Menambahkan pelanggan ke antrian

for (int i = 1; i <= jumlahPelanggan; ++i) {

Pelanggan p;

p.nomorAntrian = i;

std::cout << "Masukkan waktu layanan untuk pelanggan " << i << " (dalam menit): ";

std::cin >> p.waktuLayanan;

antrian.enqueue(p);

}

// Melakukan simulasi antrian

simulasiAntrian(antrian, jumlahLoket);

// Menampilkan statistik setelah simulasi selesai

tampilkanStatistik(totalWaktuLayanan, pelangganTerlayani, antrian.isEmpty() ? 0 : jumlahPelanggan - pelangganTerlayani);

return 0;

}

```

Dalam program ini:

- Antrian pelanggan ditambahkan menggunakan `enqueue` dan dilayani sesuai urutan dengan `dequeue`.

- Waktu layanan dihitung di setiap loket, dan jika waktu layanan habis, loket akan melayani pelanggan berikutnya dalam antrian.

- Statistik seperti jumlah pelanggan terlayani, rata-rata waktu tunggu, dan sisa antrian ditampilkan setelah simulasi.

Jadi struktur ini, program simulasi antrian layanan pelanggan di bank dapat mengelola dan menyajikan data statistik secara efisien dan akurat.