

Soal Tugas Besar 1 IF 2210 – OOP

Peraturan Umum

1. Tugas Besar 1 dikerjakan dalam kelompok, dengan masing-masing kelompok beranggotakan 2 (dua) orang.
2. Tugas Besar 1 dikerjakan dengan menggunakan bahasa **C++**,
3. Diperbolehkan untuk menggunakan seluruh STL yang terdapat dalam bahasa **C++**,
4. Praktikan harus membuat *test case* dan deskripsinya, untuk setiap kasus uji,
5. Program harus berstruktur modular sehingga perubahan dapat dilakukan dengan mudah dan cepat. **Menjelang pengumpulan mungkin akan ada permintaan perubahan mendadak.**

Deskripsi Soal

Berikut adalah deskripsi dari soal tugas besar ini :

Pada sebuah bank, ada N teller, di depan masing-masing teller ada antrian. Ada N buah teller dinomori T(0) s.d T(N-1). Q(i) adalah antrian di depan teller T(i).

T(i) statusnya 0: menganggur/idle atau 1: sedang melayani

Program utama akan memproses sederetan “event” yang diberikan sebagai input, dan dipastikan eventnya teratur waktu dan membesar (mencerminkan kejadian sesuai dengan berjalannya waktu). Jika ada event yang waktunya bersamaan, anda harus memproses sesuai urutan input. Sebuah event adalah type yang terdiri dari tiga komponen : <T: DateTime; Kode:char; i:integer> yang dijelaskan sebagai berikut :

1. T adalah type DateTime, dengan Date dan Time yang harus dibuat sendiri, dengan method yang hanya diperlukan untuk persoalan. Format Input DateTime : **DD-MM-YYYY;HH:MM:SS**
2. Kode adalah sebuah karakter yang bernilai ‘A’ atau ‘D’. A = Arrival (kedatangan pelanggan) dan D = Departure, seorang pelanggan selesai dilayani sehingga harus dihapus dari Queue.
3. i adalah nomor ID pelanggan (di-*generate* secara otomatis oleh program anda teratur mulai dari 1 pada saat Arrival).

Program anda akan memproses deretan event yang diberikan sesuai urutan pembacaan secara sekuensial, dan akan berhenti jika $T \geq T_{max}$, yang merupakan jam tutup teller.

Jika T_{max} tercapai, program harus berhenti menangani deretan event, dan akan melakukan penghapusan ke semua pelanggan yang sedang tersisa dengan “merata” artinya ulangi hapus satu per satu dari Q[1], Q[2],...Q[N]. Merata artinya bukan menghabiskan sebuah Queue sampai kosong sekaligus, tapi digilir penghapusannya.

Jockeying

Fenomena “jockeying” dalam sebuah antrian adalah terjadinya seorang pelanggan pindah ke antrian lain, karena sesuatu sebab. Yang paling sering adalah karena melihat bahwa Queue di dekatnya lebih pendek. Padahal, belum tentu kalau antrian lebih pendek akan lebih cepat dilayani sebab tergantung kepada lamanya pelayanan pelanggan. Fenomena jockeying dapat menyebabkan pelanggan hanya berpindah-pindah antrian tapi malahan tidak terlayani. Pada Tugas Besar ini, anda akan membuat sebuah algoritma

simulasi jockeying ke antrian ke-j ketika terjadi departure di sebuah antrian ke-i karena banyaknya pelanggan yang mengantri di j menjadi lebih kecil dari banyaknya yang mengantri di antrian ke-i.

Spesifikasi “jockeying” adalah sebagai berikut :

```
int Jockeying(int iOrigin)
/*
iOrigin = nomor Queue asal

Fungsi Jockeying menghasilkan j yaitu nomor Queue tujuan (jika terjadi jockeying), atau
-1 (jika tidak terjadi jockeying)

Syarat terjadinya jockeying :
1. Ada queue lain yang lebih pendek dengan selisih lebih dari 2 elemen

Mensimulasikan pelanggan yang berpindah dari Q[iOrigin] ke Q[j] (jika ada)., dengan j
!= iOrigin dan NbElmt(j) paling minimum.

Jika terdapat lebih dari satu Q[j] dengan NbElmt(j) minimum, pilih j yang paling dekat
dengan iOrigin dan j yang memiliki indeks lebih kecil.
*/
```

Contoh kasus jockeying adalah sebagai berikut :

```
Kasus - 1
Q[0] = {1,4,7,10,13}
Q[1] = {11,14}
Q[2] = {12,15}
Jika terjadi departure dari Q[0], Q[1], ataupun Q[2], tidak ada jockeying.
```

```
Kasus - 2
Q[0] = {13,17}
Q[1] = {2,6,10,14,18}
Q[2] = {11,15,19}
Q[3] = {20}
Jika terjadi departure dari Q[0], Q[2], ataupun Q[3], tidak ada jockeying.
Jika terjadi departure dari Q[1], maka akan terjadi jockeying oleh elemen (7) dari
Q[1] menuju Q[3].
```

```
Kasus - 3
Q[0] = {17,21}
Q[1] = {2,6,10,14,18,22}
Q[2] = {15,19,23}
Q[3] = {20,24}
Jika terjadi departure dari Q[0], Q[2], ataupun Q[3], tidak ada jockeying.
Jika terjadi departure dari Q[1], maka akan terjadi jockeying oleh elemen (13) dari
Q[1] menuju Q[0].
```

Program Utama

Program utama akan melakukan hal-hal berikut :

1. Meminta input TMax,
2. Meminta input jumlah teller,
3. Mengulangi membaca input "event" (mungkin mengandung data dengan baris terakhir < TMax, atau data terakhir persis sama dengan Tmax, atau masih data lainnya melebihi Tmax [yang harus diabaikan],
4. Menggambarkan kondisi antrian setelah setiap input "event". Antrian yang ditampilkan hanya antrian yang tidak kosong.

Versi Pengumpulan Program

Versi-I dengan kelas antrian (Queue) dibuat sendiri

1. Versi-1.a : simulasi antrian tanpa jockeying
2. Versi-1.b : simulai antrian dengan jockeying

Versi-2 dengan kelas antrian (Queue) memakai STL

1. Versi-2.a : simulasi antrian tanpa jockeying
2. Versi-2.b : simulasi antrian dengan jockeying

Contoh format Input-Output :

```
25-03-2014;18:00:00
3
25-03-2014;08:00:00 A
Q[0] = {1}
25-03-2014;12:00:00 A
Q[0] = {1}
Q[1] = {2}
25-03-2014;13:00:00 A
Q[0] = {1}
Q[1] = {2}
Q[2] = {3}
25-03-2014;14:00:00 D 1
Q[1] = {2}
Q[2] = {3}
25-03-2014;15:00:00 D 2
Q[2] = {3}
25-03-2014;18:00:00 A
Q[0] = {4}
Q[2] = {3}
Loop terminate
Departure 4
Departure 3
```

Tahapan Pengerjaan Tugas Besar

Pengerjaan Tugas Besar 1 IF 2210 akan dibagi menjadi beberapa tahap, yaitu :

1. 6 Maret 2014 : Cek spesifikasi soal dengan asisten / dosen melalui forum (semua tanya jawab harus terekam), dan merencanakan sketsa kelas yang harus dibuat.
2. 7 Maret 2014 – 9 Maret 2014 : Implementasi dan refinement rancangan yang telah dibuat.
3. 10 Maret – 14 Maret 2014 : fokus UTS, tugas boleh dipending
4. 15 Maret 2014 – 2 April 2014 : Implementasi dari rancangan yang telah dibuat, dan mengekstraksi dokumentasi dengan doxygen serta diagram kelas.
5. 27 Maret 2014 hadir di lab untuk pertemuan dengan asisten.
6. 2 April 2014 submisi source code dan dokumentasi.
7. 3 April 2014 : Demo Tugas Besar dan membawa deliverables :
 - a. Dokumentasi diagram kelas dan hasil ekstraksi dengan Doxygen
 - b. Log activity (kegiatan pengerjaan) sesuai dengan form yang dapat didownload di Oddy

Autograder akan menguji :

1. Kelas Queue dengan driver
2. kelas Event
3. kelas Date, Time dan DateTime
4. main program

Formulir Log Activity

Kelompok :
Anggota : 1.
2.

No	Tanggal	PK... (mulai)	Pk.... (selesai)	Aktivitas	Hasil

Total waktu pengerjaan :jam

Disarankan agar mahasiswa memanfaatkan kode ADT yang pernah dibuat dalam kuliah Struktur Data.
Kelas yang di-reuse dari tugas sebelumnya (milik sendiri)

Lesson learned dari pengerjaan tugas ini :

Log Activity ini dibuat dengan sebenarnya

Tandatangan anggota kelompok

Evaluation Form

Source Code

No	Kelas	Reuse (%)		Eksekusi		Keterangan
		Mahasiswa	Asisten	Mahasiswa	Asisten	
1	Queue					
2	Date					
3	Time					
4	DateTime					
5	Event					
6	Main					
7	Kelas					
8	Kelas....					

Dokumentasi

No	Dokumen	Kelengkapan	Kerapian	Correctness	Keterangan
	Doxygen				
	Diagram Kelas Sketsa Dikumpulkan 6 Maret 2014				
	Diagram Kelas final (reverse dengan tools)				
	Test Case	(coverage)	N/A		
	Log ACtivity				