# TUGAS PENDAHULUAN MODUL 8 ALGORITMA DAN STRUKTUR DATA



# PERATURAN TP

- Jawaban TP dikerjakan di buku khusus TP ASD
- Buku TP ASD ukuran B5; kelas yang dicantumkan di buku TP adalah kelas tempat mengikuti praktikum.
- Jawaban HARUS DITULIS TANGAN dan HARUS JELAS TERBACA. Serta melampirkan PRINTSCREEN hasil program.
- Dikumpulkan maksimal Senin, 9 November 2015 pukul 09.00 WLABIF.
- Keterlambatan pengumpulan : Nilai TP dan Tes Awal = 0 (tapi masih boleh mengikuti praktikum).
- TP MERUPAKAN SYARAT MASUK PRAKTIKUM
- TP dikumpulkan di **LOKER TP ASD** sesuai kelas masing-masing.
- Asisten **TIDAK BERTANGGUNG JAWAB** terhadap TP yang tidak sesuai **LOKER KELAS** masing-masing.
- SEGALA BENTUK PLAGIAT BENAR-BENAR TIDAK ADA TOLERANSI.
- Asisten praktikum berhak **MENOLAK TP** yang tidak sesuai ketentuan.

**SELAMAT MENGERJAKAN ©** 

- 1. buatlah ADT queue menggunakan single linked list sbb:
  - ADT elemen queue menggunakan single linked list dengan infotype integer
  - buatlah fungsi dan prosedur berikut
    - prosedur CreateQueue( i/o: Q : queue )
    - o fungsi alokasi(x:infotype):address
    - prosedur add( i/o: Q : queue, i: P : address )
    - o fungsi del( i/o: Q : queue ) : address
    - prosedur printInfo( i: Q : queue )

## 2. buatlah program untuk mensimulasikan suatu antrian

- Diilustrasikan antrian pelanggan di toko yang menunggu untuk dilayani oleh kasir.
- Menunggu adalah elemen penting dari antrian.
- Tujuan dari sistem antrian adalah dengan memanfaatkan server (teller, CPU, operator, dan sebagainya) semaksimal mungkin, sambil menjaga waktu tunggu dalam batas yang wajar.
- Dengan adanya simulasi antrian, pihak toko dapat memperkirakan berapa banyak kasir yang harus disediakan untuk dapat melayani setiap pelanggan dalam waktu tunggu yang "wajar", tapi tidak memiliki terlalu banyak kasir melebihi jumlah pelanggan
- Sebuah perusahaan dapat menentukan kompromi paling optimal antara jumlah server dan waktu tunggu

### ilustrasi kasus:

- diilustrasikan 1 buah antrian pelanggan toko yang siap dilayani
- waktu pelayanan setiap pelanggan rata-rata adalah 1 menit
- dalam satu menit rata-rata ada 0-2 pelanggan baru yang masuk ke dalam antrian

### tugas:

simulasikan alur pelanggan yang menunggu dan dilayani kasir dalam n menit, sehingga dalam n menit tersebut dapat dihitung :

- berapa banyak pelanggan yang dilayani
- rata-rata waktu tunggu pelanggan di antrian
- waktu tunggu terlama yang terjadi di antrian

Untuk mensimulasikan kasus di atas, buatlah fungsi dan prosedur berikut

### [gunakan ADT pada nomor 1]

### **ASUMSI**

- elemen queue merepresentasikan pelanggan
- infotype elemen queue bertipe integer yang menandakan pada menit ke berapa pelanggan tersebut masuk

# prosedur randomPelanggan( i/o : Q : queue, i : n : integer )

```
prosedur akan merandom sebuah angka k dalam rentang [0-3] jika k=1, maka tambahkan 1 pelanggan ke dalam queue dengan info = n jika k=2, maka tambahkan 2 pelanggan ke dalam queue dengan info = n jika k=0 atau k=3 maka tidak dilakukan apa-apa
```

# fungsi layaniPelanggan( i/o : Q : queue, i : n : integer ) : integer

```
jika Q tidak kosong, maka kasir akan melayani antrian terdepan remove pelanggan terdepan p = del(Q) fungsi akan mengembalikan integer waktu tunggu pelanggan yang dilayani n adalah menit waktu dilayani waktu tunggu = n - info(p)
```

jika Q kosong, maka fungsi akan mengembalikan nilai integer -1 yang menandakan tidak ada yang dilayani

# prosedur simulasi( total\_menit : integer )

- prosedur akan menginisialisasi queue dan mensimulasikan random pelanggan masuk dan pelanggan yang dilayani setiap menitnya hingga total\_menit
- prosedur akan melayani pelanggan di dalam antrian dan menyimpan waktu tunggu nya, dan menghitung total pelanggan yang telah dilayani
- di akhir prosedur akan ditampilkan total pelanggan yang dilayani, rata-rata waktu tunggu pelanggan, dan waktu tunggu terlama oleh pelanggan

```
Algoritma
createQueue(Q)
menit <- 0
total_dilayani <- 0
total_waktu_tunggu <- 0
waktu_tunggu_terlama <- 0

while (menit < total_menit) do
printInfo(Q)
waktu_tunggu <- layaniPelanggan(Q, menit)
jika waktu_tunggu >= 0 maka
total_dilayani ditambah satu
waktu_tunggu akan ditambahkan kepada total_waktu_tunggu
cek apakah waktu_tunggu > waktu_tunggu_terlama
randomPelanggan(Q, menit)
menit <- menit + 1
```

jalankanlah Simulasi tersebut

# Ilustrasi hasil simulasi

Contoh disimulasikan dengan total\_menit = 5
 simulasi( 5 )

menit	Layani Pelanggan	Rando m k	Pelangan masuk	Ilustrasi diagram
0	Queue kosong			<u>-</u> -
		1	Enqueue(0)	0
1	dequeue			<u>-</u>
	Dilayani = 1 W Tunggu = 1 Total tunggu = 1	2	Enqueue(1) Enqueue(1)	1 4 1 4 4
2	dequeue			1 4 4
	Dilayani = 2 W Tunggu = 1 Total tunggu = 2	2	Enqueue(2) Enqueue(2)	1 2 2 2
3	dequeue			2 2 2
	Dilayani = 3 W Tunggu = 2 Total tunggu = 4	1	Enqueue(3)	2 2 3 4
4	dequeue			2 3 4 4
	Dilayani = 4 W Tunggu = 2 Total tunggu = 6	0		2 3 4
	Data rata tur	_ 1 F monit		
Rata-rata tunggu = 1.5 menit Max tunggu = 2				