

**Topik: ADT Queue**

**Tujuan Praktikum:**

- Memahami konsep dan penggunaan ADT Queue.
- Memahami implementasinya dalam bahasa C.

**PETUNJUK PRAKTIKUM:**

1. Setiap ADT dibuat dengan format penamaan file sebagai berikut:
  - a. Untuk file header: P07<nim><noSoal>.h
  - b. Untuk file realisasi: P07<nim><noSoal>.c
  - c. Untuk file driver: P07<nim>m<noSoal>.c

Dengan:

- <nim> : NIM peserta.
- <noSoal> : nomor soal.

Contoh hasil kerja mahasiswa dengan NIM 13510500 untuk soal 1:  
P07135105001.h; P07135105001.c; P0713510500m1.c

2. Untuk setiap file yang Anda buat, cantumkan header sebagai berikut:

<pre>/* NIM&gt;Nama : Nama file : Topik : Tanggal : Deskripsi :</pre>	<pre>*/</pre>
---	---------------

3. Seluruh file di-upload setelah dikompres menjadi 1 file dengan nama: P07<nim>.zip.
4. Softcopy materi kuliah/diktat, termasuk yang terkait dengan pemrograman dengan Bahasa C dapat dilihat pada situs <http://kuliah.itb.ac.id/app245/course/> pada link **IF2110 Algorithm & Data Structure**.
5. HANYA ADT YANG DAPAT DI-COMPILE YANG AKAN DIPERIKSA. File yang tidak dapat di-*compile* akan otomatis mendapatkan nilai 0.
6. Tugas ini bersifat INDIVIDUAL. Anda dipersilakan membaca dan menggunakan material yang Anda punya, tetapi tidak ada toleransi bagi pencontek. Jika terbukti, baik yang dicontek maupun yang mencontek akan mendapatkan nilai 0.
7. Ikuti petunjuk asisten untuk pengumpulan tugas ini.

**SELAMAT BEKERJA.**

### Soal 1. Simulasi antrian bank

Anda diminta untuk membuat sebuah program simulasi sederhana yang menggambarkan antrian pengunjung di suatu bank.

- a. Ambillah ADT Queue dengan Representasi Tabel Kontigu Alternatif III dengan Alokasi Memori Dinamik yang telah Anda kerjakan sebagai tugas pra praktikum. Queue akan digunakan untuk mensimulasikan antrian pengunjung bank yang akan dilayani. Setiap elemen queue menyatakan waktu kedatangan pengunjung (integer > 0).
- b. Tambahkan prosedur PrintQueue berikut ini ke ADT tersebut:

```
procedure PrintQueue (input Q : Queue)
{ I.S. : Q terdefinisi, mungkin kosong }
{ F.S. : Semua elemen Q tertulis di layar, dengan format:
  [ el-head ... el-tail ], misalnya [ 4 8 12 9 10 ].
  Jika kosong, tampilan di layar adalah []. }
```

- c. Buatlah sebuah program yang memanfaatkan ADT Queue tersebut dan melakukan hal-hal sebagai berikut:

1. Membaca sebuah integer positif yang merepresentasikan ukuran Queue, misalnya  $n$ . Diasumsikan masukan ukuran ini selalu benar (integer > 0).
2. Mendefinisikan sebuah queue, misalnya  $Q$ , dengan size  $n$ .
3. Mengelola waktu sistem, misalnya  $t$ , yang mencatat waktu sistem *current*. Pada saat awal,  $t$  diinisialisasi 0.
4. Pengguna memasukkan sejumlah pasangan nilai <waktu> <opr> dengan:

- **waktu** adalah waktu munculnya event yang akan ditangani. Lakukan validasi agar bernilai lebih besar atau sama dengan  $t$ .
- **opr** adalah sebuah string yang menyatakan kode event/kejadian yang harus ditangani. Hanya ada 3 nilai yang mungkin untuk **opr**, yaitu: “add” yang menyatakan kedatangan seorang pengunjung ke bank dan masuk ke antrian, “del” yang menyatakan seorang pengunjung keluar dari antrian untuk dilayani, dan “close” yang menyatakan bahwa bank ditutup. Asumsi masukan opr dari pengguna selalu benar.

Misalnya: masukan **24 add** menyatakan bahwa pada waktu ke-24 ada seorang pengunjung yang datang ke bank dan masuk ke antrian.

Setiap kali menerima masukan, program akan mengeset  $t$  dengan nilai <waktu> dan kemudian melanjutkan eksekusi sesuai **opr** yang diterima:

- jika “add”, elemen dengan nilai  $t$  (sesuai waktu kedatangan) akan ditambahkan ke  $Q$ .
  - jika “del”, sebuah elemen akan dihapus/keluar dari antrian. Hitung waktu tunggu/mengantri pengunjung tersebut ( $t$  – waktu kedatangan). Tampilkan  $t$ , waktu kedatangan, dan waktu tunggu pelanggan tersebut.
  - jika “close”, artinya bank tutup dan tidak boleh lagi menerima pengunjung baru (opr add tidak dapat dieksekusi lagi). Tampilkan pesan bahwa bank sudah ditutup berikut jumlah pengunjung yang masih harus diselesaikan prosesnya (ada di dalam antrian).
5. Lakukan proses 4 hingga bank ditutup dan semua pengunjung selesai dilayani (antrian kembali kosong).
  6. Di akhir simulasi, tampilkan data dan statistik berikut ini:
    - a. Waktu bank tutup dan waktu simulasi berakhir.
    - b. Jumlah pengunjung yang dilayani.
    - c. Waktu tunggu rata-rata dari seluruh pengunjung (total waktu tunggu dari masing-masing pengunjung dibagi jumlah pengunjung).
    - d. Panjang antrian rata-rata, hanya selama periode waktu bank buka.

Dalam proses menambahkan/menghapus elemen  $Q$ , perhatikan pemrosesan terhadap *queue* kosong dan *queue* penuh. Jika ada penghapusan elemen pada *queue* kosong atau

penambahan elemen pada *queue* penuh, tampilkan pesan kesalahan dan abaikan operasi tersebut, namun *t* tetap diubah ke waktu yang baru dimasukkan oleh pengguna.

Contoh interaksi program (yang dicetak tebal dan diberi garis bawah adalah masukan pengguna) dapat dilihat berikut ini. Tampilan tidak harus persis sama, namun kandungan informasi yang ditayangkan harus setara.

### Contoh 1

```
Ukuran queue: 10
Masukkan perintah: 0 add
    Q = [0]
Masukkan perintah: 0 del
    waktu kedatangan = 0, waktu tunggu = 0, Q = []
Masukkan perintah: 2 add
    Q = [2]
Masukkan perintah: 2 add
    Q = [2 2]
Masukkan perintah: 5 del
    waktu kedatangan = 2, waktu tunggu = 3, Q = [2]
Masukkan perintah: 9 add
    Q = [2 9]
Masukkan perintah: 10 del
    waktu kedatangan = 2, waktu tunggu = 8, Q = [9]
Masukkan perintah: 12 add
    Q = [9 12]
Masukkan perintah: 14 add
    Q = [9 12 14]
Masukkan perintah: 14 del
    waktu kedatangan = 9, waktu tunggu = 5, Q = [12 14]
Masukkan perintah: 15 add
    Q = [12 14 15]
Masukkan perintah: 17 close
    Bank tutup, masih ada 3 pengunjung tersisa
Masukkan perintah: 18 del
    waktu kedatangan = 12, waktu tunggu = 6, Q = [14 15]
Masukkan perintah: 19 del
    waktu kedatangan = 14, waktu tunggu = 5, Q = [15]
Masukkan perintah: 23 del
    waktu kedatangan = 15, waktu tunggu = 8, Q = []

Bank tutup pada t = 17, layanan selesai pada t = 23
Jumlah pengunjung dilayani: 7 orang
Waktu tunggu rata-rata: 5 satuan waktu
Panjang antrian rata-rata: 1.647 orang
```

Keterangan: cara perhitungan panjang antrian rata-rata:

$(0 \text{ orang} * 2 \text{ satuan waktu} + 1 \text{ orang} * 6 \text{ satuan waktu} + 2 \text{ orang} * 7 \text{ satuan waktu} + 3 \text{ orang} * 2 \text{ satuan waktu}) / 17 \text{ satuan waktu} = 1.647 \text{ orang}$

### Contoh 2

Ukuran queue: 3  
Masukkan perintah: 2 del  
Queue kosong, tidak ada pengunjung yang akan dilayani  
Masukkan perintah: 5 add  
Q = [5]  
Masukkan perintah: 7 add  
Q = [5 7]  
Masukkan perintah: 6 del  
Waktu yang dimasukkan tidak valid, seharusnya  $\geq 7$   
Masukkan perintah: 8 add  
Q = [5 7 8]  
Masukkan perintah: 10 add  
Queue penuh, tidak bisa menerima pengunjung baru  
Masukkan perintah: 12 del  
waktu kedatangan = 5, waktu tunggu = 7, Q = [7 8]  
Masukkan perintah: 13 del  
waktu kedatangan = 7, waktu tunggu = 6, Q = [8]  
Masukkan perintah: 16 del  
waktu kedatangan = 8, waktu tunggu = 8, Q = []  
Masukkan perintah: 17 del  
Queue kosong, tidak ada pengunjung yang akan dilayani  
Masukkan perintah: 16 close  
Waktu yang dimasukkan tidak valid, seharusnya  $\geq 17$   
Masukkan perintah: 18 close  
Bank tutup, tidak ada pengunjung tersisa  
  
Bank tutup pada t = 18, layanan selesai pada t = 18  
Jumlah pengunjung dilayani: 3 orang  
Waktu tunggu rata-rata: 7 satuan waktu  
Panjang antrian rata-rata: 1.167 orang

Keterangan: cara perhitungan panjang antrian rata-rata:

$$(0 * 7 + 1 * 5 + 2 * 2 + 3 * 4) / 18 = 1.167$$

### Contoh 3

Ukuran queue: 5  
Masukkan perintah: 2 del  
Queue kosong, tidak ada pengunjung yang akan dilayani  
Masukkan perintah: 10 close  
Bank tutup, tidak ada pengunjung tersisa  
  
Bank tutup pada t = 10, layanan selesai pada t = 10  
Tidak ada pengunjung yang dilayani