

Topik: ADT Stack

Tujuan Praktikum:

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

PETUNJUK PRAKTIKUM:

1. Setiap ADT dibuat dengan format penamaan file sebagai berikut:
 - a. Untuk file header: P06<nim><noSoal>.h
 - b. Untuk file realisasi: P06<nim><noSoal>.c
 - c. Untuk file driver: P06<nim>m<noSoal>.c

Dengan:

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

Contoh hasil kerja mahasiswa dengan NIM 13510500 untuk soal 1:
P06135105001.h; P06135105001.c; P0613510500m1.c

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

```
/* NIM>Nama :  
   Nama file :  
   Topik    :  
   Tanggal  :  
   Deskripsi :  
                                                    */
```

3. Seluruh file di-upload setelah dikompres menjadi 1 file dengan nama: P06<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. Cetak dan Filter Stack (Bobot : 50%)

- Ambillah ADT Stack dengan alokasi memori **statik** yang telah Anda kerjakan sebagai tugas pra-praktikum.
- Buatlah sebuah program utama, yang melakukan hal-hal sebagai berikut:
 - Membaca sejumlah bilangan dari keyboard dan memasukkannya ke dalam suatu stack, misalnya S. Pembacaan diakhiri jika salah satu dari dua kondisi berikut terpenuhi:
 - Dimasukkannya angka 999 yang menandai akhir dari pemasukan angka (angka 999 tidak dimasukkan ke dalam stack).
 - Jika dari awal pengguna memasukkan angka 999, S menjadi stack kosong.
 - Kapasitas S sudah mencapai maksimum.
 - Menuliskan isi S mulai dari “top” ke “bottom” secara horizontal (ke samping, lihat contoh) dengan hanya memanfaatkan operasi-operasi dasar stack (push dan pop). Di akhir penulisan, S harus kembali seperti kondisi semula.
 - Memfilter S dengan cara membuang semua elemen bilangan ganjil. Stack yang kosong tidak bisa difilter. Jika S kosong, tuliskan pesan: “Tidak dapat difilter”.
 - Menuliskan kembali isi S sebagaimana yang dilakukan pada butir 2.Untuk menyelesaikan program di atas, tambahkanlah dua buah prosedur berikut ini dalam program utama Anda dan buatlah realisasinya:

```
void TulisStack (Stack S);  
/* Menuliskan isi S ke layar dari top ke bottom secara horizontal (ke samping, */  
/* lihat contoh) dengan hanya memanfaatkan operasi-operasi dasar stack */  
/* I.S. S terdefinisi, mungkin kosong */  
/* F.S. Isi S tertulis di layar dari top ke bottom. Di akhir penulisan, S */  
/* kembali ke kondisi I.S. */  
/* Jika S kosong, tuliskan “Stack kosong” */  
void FilterStack (Stack * S);  
/* I.S. Stack S terdefinisi dan tidak kosong */  
/* F.S. S hanya berisi bilangan genap dan 0, semua bilangan ganjil dibuang */  
/* Urutan elemen S sesuai urutan semula S, tanpa elemen bilangan ganjil */
```

Jika perlu menambahkan fungsi/prosedur lain, buatlah definisi, spesifikasi, dan realisasinya.

Contoh interaksi program (yang dicetak tebal dan diberi garis bawah adalah masukan pengguna):

Contoh-1	Contoh-2
Masukkan isi stack: <u>10</u> <u>3</u> <u>19</u> <u>14</u> <u>15</u> <u>8</u> <u>25</u> <u>22</u> <u>1</u> <u>6</u> {Ketr.: berhenti karena stack penuh} 6 1 22 25 8 15 14 19 3 10 Setelah dibuang bilangan ganjil: 6 22 8 14 10	Masukkan isi stack: <u>0</u> <u>1</u> <u>2</u> <u>3</u> <u>999</u> 0 1 2 3 Setelah dibuang bilangan ganjil: 0 2

Contoh-3 Masukkan isi stack: <u>5</u> <u>15</u> <u>25</u> <u>35</u> <u>45</u> <u>999</u> 5 15 25 35 45 Setelah dibuang bilangan ganjil: Stack kosong	Contoh-4 Masukkan isi stack: <u>999</u> Stack kosong Tidak dapat difilter
---	--

Soal 2. Evaluasi Ekspresi Postfix (Bobot : 50%)

Dengan memanfaatkan ADT Stack dengan alokasi memori **statik** yang telah Anda kerjakan sebagai tugas pra-praktikum dan Mesin Token yang dihasilkan pada praktikum 05, buatlah sebuah program yang membaca sebuah pita karakter yang hanya berisi bilangan positif dan 0 serta operator '+', '-', '*', '/', dan '^' yang membentuk sebuah ekspresi aritmatika dalam notasi postfix dan mengevaluasi ekspresi yang terdapat pada pita karakter tersebut.

Secara lebih rinci, program memproses hasil dari mesin token sebagai berikut:

- Apabila menerima token bilangan, maka bilangan yang dibaca dimasukkan ke dalam stack. Kemudian isi stack ditampilkan ke layar, mulai dari elemen top (memanfaatkan prosedur TulisStack yang dapat dicopy dari program yang dihasilkan di soal 1).
- Apabila menerima token operator, maka operan yang sesuai diambil dari stack, sub ekspresi yang harus dieksekusi dituliskan ke layar (format <operan1> <operator> <operan2>), hitung hasil sub ekspresi tersebut dan masukkan ke dalam stack, kemudian tampilkan isi stack ke layar.
- Apabila pembacaan pita karakter telah selesai, hasil akhir ekspresi dituliskan ke layar.

Program akan menuliskan pesan "Ekspresi kosong" apabila pita karakter kosong.

Contoh masukan dan keluaran program:

<p>Isi pita karakter: 25 10 * 15 +.</p> <p>Tampilan di layar: 25 10 25 25 * 10 250 15 250 250 + 15 265 Hasil ekspresi: 265</p>	<p>Isi pita karakter: 4 5 * 3 10 * 1 1 + / +.</p> <p>Tampilan di layar: 4 5 4 4 * 5 20 3 20 10 3 20 3 * 10 30 20 1 30 20 1 1 30 20 1 + 1 2 30 20 30 / 2 15 20 20 + 15 35 Hasil ekspresi: 35</p>
<p>Isi pita karakter: 25 .</p> <p>Tampilan di layar: 25 Hasil ekspresi: 25</p>	<p>Isi pita karakter: .</p> <p>Tampilan di layar: Ekspresi kosong</p>