

## Topik: ADT Array

### Tujuan Praktikum:

- Memahami konsep Array.
- Memahami implementasinya dalam bahasa C.

### PETUNJUK PRAKTIKUM:

1. Setiap ADT dibuat dengan format penamaan file sebagai berikut:
  - a. Untuk file header: P03<nim><noSoal>.h
  - b. Untuk file realisasi: P03<nim><noSoal>.c
  - c. Untuk file driver: P03<nim>m<noSoal>.c

Dengan:

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

Contoh hasil kerja mahasiswa dengan NIM 13510500 untuk soal 1:  
P03135105001.h; P03135105001.c; P0313510500m1.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: P03<nim>.zip.
4. Softcopy materi kuliah/diklat, 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. ADT Array

#### ADT Array dengan Elemen Terurut Membesar, Representasi Eksplisit – Statik

Ambillah ADT array dengan representasi **eksplisit** dan alokasi memori **statik** yang telah Anda kerjakan sebagai tugas pra-praktikum. Anda ditugaskan untuk melakukan modifikasi ADT array tersebut menjadi **ADT array dengan elemen terurut membesar**. Modifikasi yang dilakukan adalah:

1. Buanglah primitif-primitif yang tidak relevan dengan ADT array dengan elemen terurut membesar, yaitu primitif-primitif yang masuk dalam kelompok:
  - Searching (searching biasa yang tidak mempertimbangkan tabel terurut membesar).
  - Nilai ekstrem (yang tidak mempertimbangkan bahwa tabel terurut membesar).
  - Operasi inverse tabel dan apakah sebuah tabel simetris.
  - Sorting (tabel sudah pasti terurut sehingga tidak perlu di-sort).
  - Menambah dan menghapus elemen (yang tidak mempertimbangkan bahwa tabel terurut membesar).
  - Kelompok primitif terkait tabel dengan elemen unik.
2. Jika Anda belum menyelesaikan primitif-primitif yang masuk kelompok untuk tabel dengan elemen terurut membesar, selesaikanlah primitif-primitif tersebut.
3. Tambahkan fungsi/prosedur di bawah ini:

```
function SearchUrutB (T : TabInt, X : ElType) → boolean
{ Prekondisi: Tabel boleh kosong. Jika tidak kosong, elemen terurut
  membesar. }
{ Mengirimkan true jika X ada di dalam T dan false jika X tidak ditemukan
  di T. }

function SumTab (T : TabInt) → integer
{ Prekondisi : Tabel T tidak kosong }
{ Mengirim hasil penjumlahan semua elemen dalam T }

procedure UpdateElmt (input/output T : TabInt, input i : IdxType, input X
: ElType)
{ I.S. : T terdefinisi, tidak kosong.
  i merupakan indeks valid dalam T.
  X terdefinisi. }
{ F.S. : Nilai elemen T pada indeks ke-i berubah menjadi X.
  T tetap terurut membesar, sehingga ada kemungkinan dilakukan
  proses pengurutan kembali. }
{ Proses : ... -> tuliskan bagaimana Anda menyelesaikannya. }

procedure DelAllX (input/output T : TabInt, input X : ElType)
{ I.S. : T terdefinisi, boleh kosong. X terdefinisi. }
{ F.S. : Semua elemen T yang bernilai X dihapus dari tabel penampung, jika
  X ada di T. Jika X tidak ada, T tetap. }
{ Proses : ... -> tuliskan bagaimana Anda menyelesaikannya. }
```

Jika perlu membuat prosedur/fungsi tambahan, jangan lupa tuliskan definisi, spesifikasi, dan realisasi dari prosedur/fungsi tersebut.

4. Buatlah sebuah program utama yang memanfaatkan ADT array dengan elemen terurut membesar dengan urutan sebagai berikut:
  - Membuat sebuah tabel kosong.
  - Menentukan suatu nilai efektif tertentu (yang harus  $> 0$  dan  $\leq \text{IdxMax}$ ) dan kemudian mengisi tabel sebanyak nilai efektif tertentu. Dalam proses pengisian, pengguna boleh memasukkan nilai secara tidak terurut, tapi hasil akhirnya tabel tetap harus terurut.
  - Menuliskan isi tabel ke layar.

- Menjumlahkan isi seluruh tabel dan menuliskan hasilnya ke layar.
- Menerima sebuah nilai X bertipe ElType dan nilai indeks i bertipe IdxType dan kemudian meng-update nilai elemen tabel pada indeks ke-i dengan nilai X serta menuliskan kembali isi tabel ke layar. Harus diperiksa apakah masukan indeks benar sesuai nilai efektif tabel.
- Menerima sebuah nilai Y bertipe ElType dan menghapus nilai Y pertama yang ditemukan dalam tabel serta menuliskan kembali isi tabel ke layar.
- Menerima sebuah nilai Z bertipe ElType dan menghapus semua nilai Z yang ditemukan dalam tabel serta menuliskan kembali isi tabel ke layar.

Contoh interaksi program utama (bagian yang digarisbawahi adalah masukan dari pengguna):

```
*** PROGRAM UTAMA UNTUK ADT ARRAY DENGAN ELEMEN TERURUT ***

*** Inisialisasi tabel ***
Masukkan nilai efektif tabel = 6
Masukkan nilai elemen ke-1 = 10
Masukkan nilai elemen ke-2 = 4
Masukkan nilai elemen ke-3 = -1
Masukkan nilai elemen ke-4 = 0
Masukkan nilai elemen ke-5 = 2
Masukkan nilai elemen ke-6 = 4
Isi tabel =
[1] -1
[2] 0
[3] 2
[4] 4
[5] 4
[6] 10

*** Penjumlahan seluruh elemen tabel ***
Hasil penjumlahan seluruh elemen tabel = 19

*** Update elemen tabel ***
Masukkan sebuah nilai baru = -3
Masukkan indeks untuk update (antara 1 s.d. 6) = 0
Masukan indeks salah.
Masukkan indeks untuk update (antara 1 s.d. 6) = 3
Isi tabel =
[1] -3
[2] -1
[3] 0
[4] 4
[5] 4
[6] 10

*** Penghapusan sebuah nilai elemen tabel ***
Masukkan nilai yang akan dihapus = -1
Isi tabel =
[1] -3
[2] 0
[3] 4
[4] 4
[5] 10

*** Penghapusan semua elemen bernilai tertentu ***
Masukkan nilai yang akan dihapus = 4
Isi tabel =
[1] -3
[2] 0
[3] 10
```