



# Praktikum Struktur Data

## Modul PSDA

### PSDA-01. ADT STACK

#### Bagian 1. Representasi Tabel Kontigu dengan Alokasi Memori Statik

1. Buatlah ADT Stack yang diimplementasikan dengan tabel kontigu dengan **alokasi memori statik** dalam bahasa C berdasarkan atau pada spesifikasi di bawah ini dalam Bahasa C dengan pembagian modul sebagaimana dijelaskan dalam kuliah. Untuk type boolean, gunakan file berisi definisi type boolean secara terpisah (**boolean.h**).
2. Buatlah driver untuk memeriksa apakah seluruh primitif yang didefinisikan telah berjalan dengan baik.

```
{ MODUL STACK }
{ Deklarasi stack yang diimplementasi dengan tabel kontigu alokasi statik }
{ dan ukuran sama }
{ TOP adalah alamat elemen puncak }

constant Nil    : integer = 0    { Nil adalah stack dengan elemen kosong }
constant MaxEl  : integer = 10

type infotype : integer
type address  : integer    { indeks tabel }

{ Contoh deklarasi variabel bertipe stack dengan ciri TOP : }
{ Versi I : dengan menyimpan tabel dan alamat top secara eksplisit}
type Stack : < T : array [1..MaxEl] of infotype, { tabel penyimpan elemen }
              TOP : address { alamat TOP : elemen puncak }
              >

{ Definisi stack S kosong : S.TOP = Nil }
{ Elemen yang dipakai menyimpan nilai Stack T[1]..T[MaxEl] }
{ Jika S adalah Stack maka akses elemen : }
{ S.T[(S.TOP)] untuk mengakses elemen TOP }
{ S.TOP adalah alamat elemen TOP }
{ Definisi akses dengan Selektor : Isilah dengan selektor yang tepat }

{ ***** Prototype ***** }

{ *** Konstruktor/Kreator *** }
procedure CreateEmpty (output S : Stack)
{ I.S. Sembarang }
{ F.S. Membuat sebuah stack S yang kosong berkapasitas MaxEl }
{ jadi indeksanya antara 1..MaxEl }
{ Ciri stack kosong : TOP bernilai Nil }

{ ***** Predikat Untuk test keadaan KOLEKSI ***** }
function IsEmpty (S : Stack) → boolean
{ Mengirim true jika Stack kosong: lihat definisi di atas }
function IsFull (S : Stack) → boolean
{ Mengirim true jika tabel penampung nilai elemen stack penuh }

{ ***** Operator Dasar Stack ***** }
procedure Push (input/output S : Stack, input X : infotype)
{ Menambahkan X sebagai elemen Stack S. }
{ I.S. S mungkin kosong, tabel penampung elemen stack TIDAK penuh }
{ F.S. X menjadi TOP yang baru, TOP bertambah 1 }
procedure Pop (input/output S : Stack, input X : infotype);
{ Menghapus X dari Stack S. }
{ I.S. S tidak kosong }
{ F.S. X adalah nilai elemen TOP yang lama, TOP berkurang 1 }
```



# Praktikum Struktur Data

## Modul PSDA

### Bagian 2. Representasi Tabel Kontigu dengan Alokasi Memori Dinamik

1. Buatlah ADT Stack yang diimplementasikan dengan tabel kontigu dengan **alokasi memori dinamik** dalam bahasa C berdasarkan atau pada spesifikasi di bawah ini dalam Bahasa C dengan pembagian modul sebagaimana dijelaskan dalam kuliah. Untuk type boolean, gunakan file berisi definisi type boolean secara terpisah (**boolean.h**).
2. Buatlah driver untuk memeriksa apakah seluruh primitif yang didefinisikan telah berjalan dengan baik.

```
{ MODUL STACK }
{ Deklarasi stack yang diimplementasi dengan tabel kontigu secara dinamik }
{ dan ukuran sama }
{ TOP adalah alamat elemen puncak }

constant Nil    : integer = 0    { Nil adalah stack dengan elemen kosong }

type infotype : integer
type address  : integer    { indeks tabel }

{ Contoh deklarasi variabel bertipe stack dengan ciri TOP : }
{ Versi I : dengan menyimpan tabel dan alamat top secara eksplisit}
type Stack : < T : ...,      { tabel penyimpan elemen, tergantung bahasa }
              TOP : address,  { alamat TOP : elemen puncak }
              Size : integer { ukuran stack } >

{ Definisi stack S kosong : S.TOP = Nil }
{ Elemen yang dipakai menyimpan nilai Stack T[1]..T[Size] }
{ Jika S adalah Stack maka akses elemen : }
{ S.T[(S.TOP)] untuk mengakses elemen TOP }
{ S.TOP adalah alamat elemen TOP }
{ S.Size adalah ukuran stack }
{ Definisi akses dengan Selektor : Isilah dengan selektor yang tepat }

{ ***** Prototype ***** }

{ *** Konstruktor/Kreator *** }
procedure CreateEmpty (output S : Stack, input Size : integer)
{ I.S. Sembarang }
{ F.S. Membuat sebuah stack S yang kosong berkapasitas Size }
{ jadi indeksnya antara 1..MaxEl }
{ Ciri stack kosong : TOP bernilai Nil }

{ *** Destruktor *** }
procedure Destruct (input/output S : Stack)
{ Destruktor: dealokasi seluruh tabel memori sekaligus }

{ ***** Predikat Untuk test keadaan KOLEKSI ***** }
function IsEmpty (S : Stack) → boolean
{ Mengirim true jika Stack kosong: lihat definisi di atas }
function IsFull (S : Stack) → boolean
{ Mengirim true jika tabel penampung nilai elemen stack penuh }

{ ***** Operator Dasar Stack ***** }
procedure Push (input/output S : Stack, input X : infotype)
{ Menambahkan X sebagai elemen Stack S. }
```



## Praktikum Struktur Data

### Modul PSDA

```
{ I.S. S mungkin kosong, tabel penampung elemen stack TIDAK penuh }
{ F.S. X menjadi TOP yang baru, TOP bertambah 1 }
procedure Pop (input/output S : Stack, input X : infotype);
{ Menghapus X dari Stack S. }
{ I.S. S tidak kosong }
{ F.S. X adalah nilai elemen TOP yang lama, TOP berkurang 1 }
```