

# TUGAS PENDAHULUAN

## MODUL 12

### PENGENALAN DAN IMPLEMENTASI STRUKTUR DATA STACK

Prepared by: Ardianto Satriawan (02-2015)

#### Pengenalan

#### *Pengertian Stack*

Pada Tugas Pendahuluan praktikum kali ini, anda tidak diminta untuk mengumpulkan program, melainkan mempersiapkan apa yang dibutuhkan dalam praktikum. **Anda tidak diperkenankan ikut praktikum apabila tidak mengerjakan tugas pendahuluan ini.** Asisten akan memeriksa pekerjaan Anda di awal praktikum.

Stack adalah struktur data bentukan yang berbentuk tumpukan. Bayangkanlah data yang akan kita pakai diperlakukan seperti tumpukan piring.



Gambar 1: Ilustrasi stack dengan piring

Satu piring mewakili satu buah data. Karena piring-piring itu berada dalam tumpukan, piring yang dapat diambil hanyalah piring yang berada di paling atas. Demikian juga apabila kita ingin menambah piring baru, hanya bisa kita letakkan di atas tumpukan.

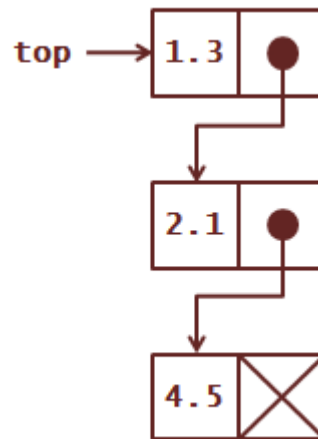
Sekarang kembali ke level abstraksi pemrograman, **stack** adalah tumpukan data. Menambahkan data hanya bisa dilakukan ke atas stack. Proses menambahkan data pada stack disebut **push**. Mengeluarkan data juga hanya bisa dilakukan untuk data yang berada di paling atas stack. Proses pengeluaran data disebut **pop**. Sebagai tambahan, data di dalam stack juga biasa disebut sebagai **element**.

Pada tugas pendahuluan ini, Anda diminta untuk mengimplementasikan tipe data bentukan untuk **stack**, fungsi **push()**, dan fungsi **pop()**. Ketiga-tiganya akan digunakan di dalam praktikum.

## Tugas 1 : Implementasi Struktur Data Element Stack

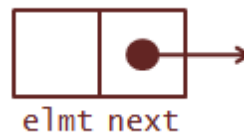
### Definisi Masalah

Pada praktikum yang akan dilaksanakan, struktur data stack diimplementasikan sebagai sebuah linked-list. Data yang akan kita gunakan sebagai “piring” dalam stack bertipe double.



Gambar 2: Ilustrasi stack berisi 3 element double diimplementasikan dengan *linked-list*

Untuk mempermudah pekerjaan kita ke depan, kita harus membuat tipe bentukan untuk **satu element dari stack**. Perhatikan ilustrasi berikut



Gambar 3: Satu element (*node*) stack

Tipe bentukan untuk satu element stack ini, kita sebut sebagai **node**. Data di dalamnya diberi nama **elmt** dan bertipe double. Pointer ke node selanjutnya diberi nama **next**. Implementasikan tipe bentukan untuk *node*.

Diberikan kode bantu sebagai berikut:

```
typedef struct node
{
    /** Fill what is missing
        in this structure
        ...
        **/
} node;
```

## Tugas 2 : Implementasi Struktur Data Stack

### *Definisi Masalah*

Setelah kita mengimplementasikan tipe bentukan untuk satu element atau *node* dari stack, maka selanjutnya Anda diminta untuk membuat tipe bentukan untuk **stack** itu sendiri. Ilustrasi dapat dilihat pada gambar 2 di atas. Pointer yang menunjuk ke elemen paling atas dari stack diberi nama **top**.

Sekali lagi, diberikan kode bantuan sebagai berikut:

```
typedef struct stacktype
{
    /** Fill what is missing
        in this structure
        ...
        ...          **/
} stacktype;
```

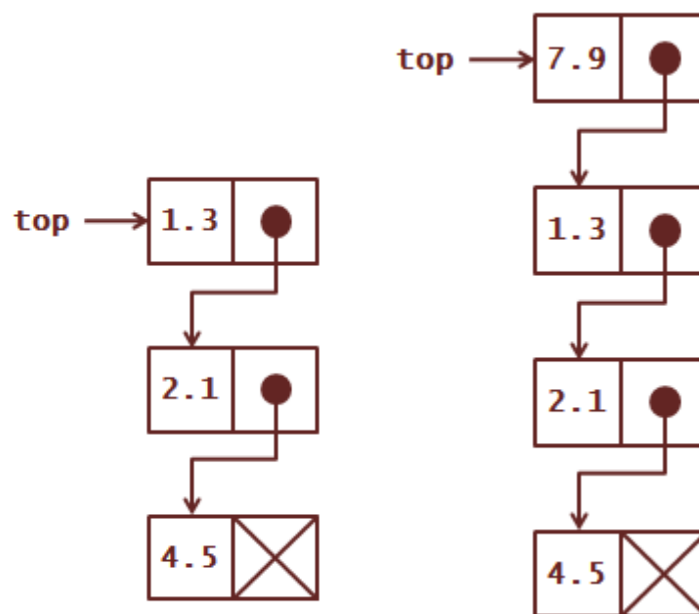
## Tugas 3 : Implementasi Fungsi Push

### *Definisi Masalah*

Prototype dari fungsi `push()` yang akan dibuat adalah sebagai berikut:

```
void push(double elmt, stacktype *stack);
```

Fungsi `push` akan menambahkan element ke atas stack. Misalkan fungsi ini dipanggil dengan `push(7.9, &stack)` saat kondisi stack seperti gambar 2, maka setelah fungsi selesai dijalankan, isi dari stack adalah sebagai berikut.



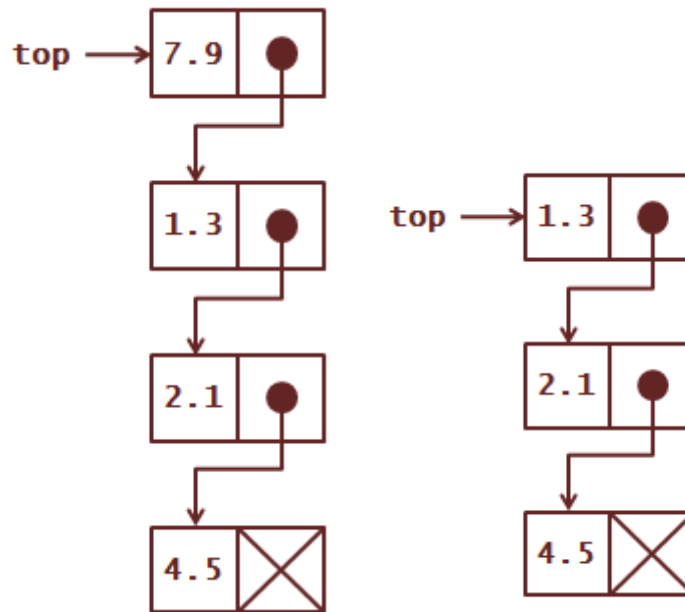
Gambar 4: Sebelum dan setelah `push(7.9, &stack)` dieksekusi

Tugas Anda adalah mengimplementasikan fungsi `push()` ini, hingga bekerja dengan baik.

#### Tugas 4 : Implementasi Fungsi Pop

#### *Definisi Masalah*

Fungsi `pop` adalah kebalikan dari fungsi `push`, berguna untuk menghilangkan elemen teratas dari `stack`.



Gambar 5: Sebelum dan setelah `pop(&stack)` dieksekusi

Namun, terdapat sedikit perbedaan antara `push` dengan `pop`. Bila pada `push`, kita tidak perlu mengembalikan (*return*) suatu nilai. Pada `pop`, dilakukan pengembalian nilai untuk data yang kita ambil. Pada ilustrasi di gambar 5, fungsi `pop` akan mengembalikan nilai 7.9.

Prototype fungsi `push` adalah:

```
double pop(stacktype *stack);
```

Tugas Anda adalah mengimplementasikan fungsi `pop` ini, hingga bekerja dengan baik.

## Petunjuk Penyerahan Tugas Pendahuluan Modul 12

Keempat tugas di atas disimpan dalam satu buah file .c, seharusnya file .c yang telah Anda buat kurang lebih seperti ini:

```
/* ... library, definition, etc */

typedef struct node
{
    /** Fill what is missing
        in this structure
        ...          **/
} node;

typedef struct stacktype
{
    /** Fill what is missing
        in this structure
        ...          **/
} stacktype;

/** function prototypes **/
void push(double elmt, stacktype *stack);
double pop(stacktype *stack);

int main(void)
{
    /** will be filled during lab session **/
}

/** function implementations **/
void push(double elmt, stacktype *stack)
{
    /** push function implementation **/
}

double pop(stacktype *stack)
{
    /** pop function implementation **/
}
```

Bawalah kode tersebut dan tunjukkan kepada asisten di awal praktikum.

**Selesai**