

## [TUGAS 2] Singly Linked List (SLL): ADT & Primitives

Buatlah ADT SLL yang akan menyimpan data sesuai dengan NIM Anda.

1. Lakukan MOD 9 terhadap digit terakhir pada NIM Anda.
2. Sisanya menentukan jenis data apa yang Anda simpan di list
  - Sisa 0 : Data buku di perpustakaan
  - Sisa 1 : Daftar tugas harian
  - Sisa 2 : Riwayat panggilan telepon
  - Sisa 3 : Data wisata dalam itinerary
  - Sisa 4: Riwayat penggunaan aplikasi
  - Sisa 5: Rantai pasokan (*supply chain*)
  - Sisa 6: Rantai kejadian dalam investigasi
  - Sisa 7: Urutan pembelajaran kursus daring
  - Sisa 8: Daftar putar musik
3. Data tersebut minimal memiliki 3 sub data. Contoh : pada data terkait daftar episode dalam serial TV, linked list bisa digunakan untuk merepresentasikan daftar episode dalam sebuah serial, dengan informasi seperti judul episode, durasi, rating episode, dan link ke episode berikutnya.

---

### LENGKAPILAH ADT SINGLY LINKED LIST BERIKUT INI BESERTA MAIN PROGRAMNYA

#### A. Isi SLL.h

- a. Deklarasikan List Anda disini

type ..... : < .....

.....

..... >

type infotype : .....

type adr : .....

type elmList : < .....

.....>

type List : <.....>

Sesuaikan dengan jenis data yang Anda dapatkan sesuai panduan di atas

- b. Tuliskan primitif Singly Linked List berdasarkan fungsi/prosedur yang ada pada file sll.cpp pada poin B

#### B. Isi SLL.cpp

- a. **procedure** Create\_List (in/out L : List)

{*I.S.* –

*F.S.* Pointer First dari List L bernilai NIL}

**Kamus data**

.....

**Algoritma**

.....  
.....

- b. **function** New\_Elemen (data : infotype) → adr  
*{Function akan membuat elemen baru berisi data dan mengembalikan pointer yang menyimpan alamat dari elemen tersebut}*

**Kamus data**

.....

**Algoritma**

.....  
.....

- c. **procedure** Insert\_First (**in/out** L : List, **In** p : adr)  
*{I.S. Terdefinisi List L yang mungkin kosong dan sebuah elemen baru yang alamatnya disimpan oleh pointer P  
F.S. Elemen baru telah masuk menjadi elemen paling pertama di list L}*

**Kamus data**

.....

**Algoritma**

.....  
.....

- d. **procedure** Insert\_Last (**in/out** L : List, **In** p : adr)  
*{I.S. Terdefinisi List L yang mungkin kosong dan sebuah elemen baru yang alamatnya disimpan oleh pointer P  
F.S. Elemen baru telah masuk menjadi elemen paling akhir di list L}*

**Kamus data**

.....

**Algoritma**

.....  
.....

- e. **procedure** Insert\_After (**in/out** L : List, **In** prec, p : adr)  
*{I.S. Terdefinisi List L yang mungkin kosong, sebuah elemen baru yang alamatnya disimpan oleh pointer P dan pointer prec  
F.S. Elemen baru telah masuk menjadi elemen setelah elemen yang alamatnya disimpan oleh pointer prec }*

**Kamus data**

.....

**Algoritma**

.....  
.....

- f. **procedure** Delete\_First (**in/out** L : List, **Out** p : adr)  
*{I.S. Terdefinisi List L yang mungkin kosong atau Cuma memiliki 1 elemen.  
F.S. Jika list kosong, maka pointer P di assign dengan NIL, jika tidak kosong maka elemen paling awal dihapus dari List L, alamatnya disimpan oleh pointer p}*

**Kamus data**

.....

### Algoritma

.....  
.....

- g. **procedure** Delete\_Last (in/out L : List, Out p : adr)  
{I.S. Terdefinisi List L yang mungkin kosong atau Cuma memiliki 1 elemen.  
F.S. Jika list kosong, maka pointer P di assign dengan NIL, jika tidak kosong maka elemen paling akhir dihapus dari List L, alamatnya disimpan oleh pointer p}

### Kamus data

.....  
.....

### Algoritma

.....  
.....

- h. **procedure** Delete\_After (in/out L : List, In prec : adr, Out p : adr)  
{I.S. Terdefinisi List L yang mungkin kosong, sebuah elemen baru yang alamatnya disimpan oleh pointer P dan pointer prec  
F.S. Elemen baru telah masuk menjadi elemen setelah elemen yang alamatnya disimpan oleh pointer prec }

### Kamus data

.....  
.....

### Algoritma

.....  
.....

- i. **procedure** Show (In L : List)  
{I.S. Terdefinisi List L yang mungkin kosong.  
F.S. Jika list kosong maka tampilkan ke layar "list kosong", jika tidak maka seluruh data pada list ditampilkan ke layar}

### C. Isi Main.cpp

{Lengkapilah titik-titik berikut ini agar pada main program :

1. Deklarasi variable List
2. Create List
3. Memasukkan 4 data ke list dengan memanfaatkan dua jenis insert
4. Memanggil show data
5. Menghapus 2 data ke list dengan memanfaatkan dua jenis delete}

//deklarasi variable List. **Nama variable List adalah huruf pertama dan huruf terakhir pada nama lengkap Anda**

.....

//create list

.....

//meminta user **data pertama** yang akan dimasukkan ke list

.....

//melakukan create new element berdasarkan data yang dimasukkan oleh user

.....

*//memanggil salah satu jenis insert (jangan insert after)*

.....

*//meminta user **data kedua** yang akan dimasukkan ke list*

.....

*//melakukan create new element berdasarkan data yang dimasukkan oleh user*

.....

*//memanggil salah satu jenis insert (berbeda dengan insert yang data pertama, jangan Insert after)*

.....

*//meminta user **data ketiga** yang akan dimasukkan ke list*

.....

*//melakukan create new element berdasarkan data yang dimasukkan oleh user*

.....

*//memanggil salah satu jenis insert (insert First)*

.....

*//meminta user **data keempat** yang akan dimasukkan ke list*

.....

*//melakukan create new element berdasarkan data yang dimasukkan oleh user*

.....

*//memanggil salah satu jenis insert (insert Last)*

.....

*//memanggil show*

.....

*//melakukan penghapusan data dengan memanfaatkan salah satu jenis delete (bukan delete after)*

.....

*//melakukan penghapusan data dengan memanfaatkan salah satu jenis delete (berbeda dengan delete sebelumnya, dan bukan delete after)*

.....

ANSWER