

## JURNAL 2 – STRUKTUR DATA

- A. Buatlah ADT Single Linked List (SLL.H dan SLL.cpp) yang akan menyimpan data **integer**. *Note: Pointer to elemen dialiaskan dengan adr (gunakan define pada c++)*
- Subprogram yang harus ada adalah:

```
1. Procedure createList(output L:list)
2. Function alokasi(data:infotype) → adr
3. Procedure insertFirst(input/output L:list, input P:adr)
4. Procedure insertLast(input/output L:list, input P:adr)
5. Procedure insertAfter(input/output L:list, input Prec,P:adr)
6. Procedure deleteFirst(input/output L:list, output P:adr)
7. Procedure deleteLast(input/output L:list, output P:adr)
8. Procedure deleteAfter(input/output L:list, input Prec:adr, output P:adr)
9. Procedure show(input L : list)
10. Function findInfo(L : list) → adr
```

Petunjuk Insert After dan Delete After:

- Prec akan berisi alamat suatu elemen list L.
- Insert After: lakukan insert pada elemen setelah Prec.
- Delete After: lakukan delete pada elemen yang ditunjuk oleh next(Prec).

- B. Buatlah main program (main.cpp) yang akan menggunakan ADT Single Linked List yang baru saja Anda buat. Pada main program, buatlah program yang akan melakukan hal dibawah ini :

1. Buatlah list kosong
2. Mintalah data angka dari user 5 angka. Setiap 1 angka akan disimpan (*insert*) ke dalam satu elemen. Kemudian tampilkan ke layar isi list setelah proses *insert*. Contoh tampilan (teks bergaris bawah adalah *input/read*):

```
Masukkan angka ke-1 : 10
Masukkan angka ke-2 : 5
Masukkan angka ke-3 : 4
Masukkan angka ke-4 : 20
Masukkan angka ke-5 : 8
Isi list saat ini adalah :
10 5 4 20 8
```

3. Jika tampilan Anda sudah sesuai di atas maka Anda berhasil melakukan proses insert data pada List.
4. Bantulah user mencari data pada list. Jika list kosong maka outputkan **Maaf List Kosong**, jika list tidak kosong, dilakukan pencarian. Jika data yang dicari ditemukan, maka outputkan **Data Ditemukan pada alamat XYZ**, jika tidak maka outputkan **"Maaf data tidak ditemukan"**.

Contoh tampilan 1 (teks bergaris bawah adalah input/read)

```
Masukkan data yang dicari : 5
DATA DITEMUKAN PADA ALAMAT 0x0?????
```

Contoh tampilan 2 (teks bergaris bawah adalah input/read)

```
Masukkan data yang dicari : 1
MAAF DATA TIDAK DITEMUKAN
```

5. Bantulah user menghapus semua data satu persatu. User diperolehkkan memilih elemen mana yang akan dihapus, elemen pertama atau elemen terakhir. Proses penghapusan selesai ketika List sudah kosong.

Contoh tampilan (teks bergaris bawah adalah input/read)

```
Kondisi awal : 10 5 4 20 8

Lokasi elemen yang dihapus(depan/belakang) ? depan
Kondisi setelah dihapus : 5 4 20 8

Lokasi elemen yang dihapus(depan/belakang) ? belakang
Kondisi setelah dihapus : 5 4 20

Lokasi elemen yang dihapus(depan/belakang) ? belakang
Kondisi setelah dihapus : 5 4

Lokasi elemen yang dihapus(depan/belakang) ? depan
Kondisi setelah dihapus : 4

Lokasi elemen yang dihapus(depan/belakang) ? belakang
Kondisi setelah dihapus : List Kosong

SELESAI
```