

TUGAS PENDAHULUAN MODUL 4
MK STRUKTUR DATA 2019/2020-2
Double Linked-List dengan pointer First dan Last

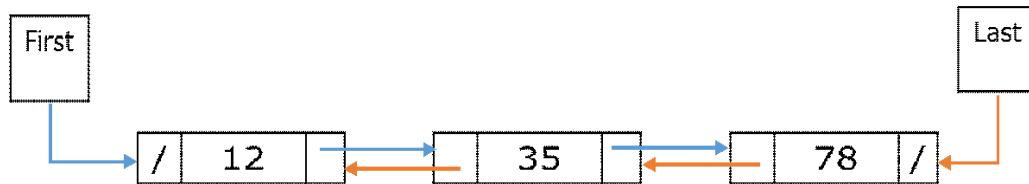
| |
|------------------------------------|
| Ketentuan Tugas Pendahuluan |
|------------------------------------|

- **JAWABAN DIKETIK DENGAN RAPIH dan SERTAKAN SCREENSHOOT CODINGAN DAN HASIL OUTPUT** untuk soal algoritma
- Jawaban diprint dan ditempel di buku tulis B5 dan sertakan identitas pada cover buku tulis.
- TP ini bersifat **WAJIB, TIDAK MENGERJAKAN BERARTI TIDAK BOLEH MENGIKUTI PRAKTIKUM.**
- **HANYA MENGUMPULKAN TETAPI TIDAK MENGERJAKAN = TIDAK BOLEH MENGIKUTI PRAKTIKUM.**
- **TIDAK BOLEH PRAKTIKUM = TIDAK DIPERKENANKAN DIDALAM RUANG PRAKTIKUM SERTA TIDAK BOLEH TAPPING RFID DAN ABSEN TANDA TANGAN.**
- **BUKU TULIS WAJIB DIBAWA SAAT PRAKTIKUM, TIDAK MEMBAWA BUKU TULIS MAKA TIDAK DIPERBOLEHKAN MENGIKUTI PRAKTIKUM**
- Deadline pengumpulan TP Modul 4 Senin, 17 Februari 2020 pukul 08.03 WIFLAB
- **TIDAK ADA TOLERANSI KETERLAMBATAN, TERLAMBAT ATAU TIDAK MENGUMPULKAN TP ONLINE MAKA DIANGGAP TIDAK MENGERJAKAN**
- **DILARANG PLAGIAT (PLAGIAT = E)**
- Kerjakan TP dengan jelas agar dapat dimengerti.
- Untuk setiap soal algoritma dibawahnya diberikan comment **NAMA** dan **NIM** seperti dibawah.
- Untuk soal algoritma, setiap nama file harus disertai dengan **NIM (Contoh : header_130416XXXX)**
- **NAMA FILE SAAT UPLOAD ONLINE : MODX_NIM_KELAS.pdf**

```
int example (int a, int b) {  
  
    /*  
    Name : Ichi Ocha  
    NIM : 1301123456  
    */  
}
```

TUGAS PENDAHULUAN

Perhatikan list berikut:



BAGIAN I: Buat ADT double linked list di atas dalam "doublelinkedlist.h".

Definisi tipe data untuk list tersebut:

```

type infotype: integer
type address: pointer to elmtList
type elmtList: <info: infotype, prev: address, next: address>
type list: <first, last: address>

function isEmpty (L: list) → boolean
{Fungsi mengirimkan true jika list kosong, dan false jika tidak kosong}

procedure createList (output L: list)
{I.S. –
F.S. terdefinisi L, list kosong}

procedure createNewElmt (input X: infotype, output P: address)
{I.S. X adalah info yang akan ditempatkan pada elemen baru yang dialokasi
F.S. terdefinisi elemen list dengan address P, dimana info dari P adalah X, atau NULL jika alokasi elemen baru gagal}

procedure insertFirst (input/output L: list, input P: address)
{I.S List L mungkin kosong. Telah terdefinisi P untuk disisipkan ke dalam L; P^.next dan p^.prev= NULL
F.S. P menjadi elemen pertama dari list L}

procedure insertAfter (input/output L: list, input/output Prec, P: address)
{I.S. Prec tidak NULL dan merupakan elemen list L. P akan disisipkan setelah Prec
F.S. P sudah disisipkan ke dalam L dan terletak setelah Prec}

procedure insertLast (input/output L: list, input P: address)
{I.S List L mungkin kosong. Telah terdefinisi P untuk disisipkan ke dalam L; P^.next dan p^.prev= NULL
F.S. P menjadi elemen terakhir pada dari list L}

procedure deleteFirst (input/output L: list, output P: address)
{I.S. L mempunyai satu atau lebih elemen
F.S. Elemen pertama sudah dihapus dan dicatat pada P. List L mungkin menjadi kosong}

procedure deleteAfter (input/output L: list, Prec: address, output P: address)
{I.S. Prec adalah elemen list L dan Prec^.next tidak NULL. Prec^.next mungkin menunjuk ke elemen terakhir
F.S. Elemen setelah Prec sudah dihapus dari L dan diingat pada P}

procedure deleteLast (input/output L: list, output P: address)
{I.S. L mempunyai satu atau lebih elemen
F.S. Elemen terakhir sudah dihapus dan dicatat pada P. List L mungkin menjadi kosong}

procedure concat (input L1, L2: list, output L3: list)
{I.S. L1 dan L2 masing-masing mempunyai satu atau lebih elemen
F.S. L3 berisi semua elemen L1 digabung dengan semua elemen L2}

function median (L: list) → real
    
```

{I.S. L mempunyai elemen terurut menaik

F.S Mencari median pada list L. List L tidak kosong. Fungsi mengembalikan median dari list L. Apabila jumlah elemen dalam list ganjil, maka median adalah nilai info dari elemen list yang posisinya di tengah, sedangkan jika genap, maka mean rata-rata nilai info dari dua elemen list yang ada di tengah}

BAGIAN 2: Buat implementasi double linked list di atas dalam “doublelinkedlist.cpp”.

BAGIAN 3: panggil semua fungsi dan prosedur di program utama “main.cpp” untuk membuktikan fungsi dan prosedur memberikan hasil yang benar.