

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 2 Senin, 03 Februari 2020 pukul 08.03 WIFLAB
- **TIDAK ADA TOLERANSI KETERLAMBATAN, TERLAMBAT ATAU TIDAK MENGUMPULKAN TP ONLINE MAKA DIANGGAP TIDAK MENGERJAKAN**
- **DILARANG PLAGIAT**
- 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 2

BACA PELAN-PELAN, KERJAKAN SECARA INDIVIDUAL. DISKUSI DIPERSILAHKAN

PRACTICE MAKE PERFECT

Selamat datang di Tugas Pendahuluan SINGLE LINKED LIST

Kali ini, melalui tugas pendahuluan, rekan-rekan akan **DIPANDU BAGAIMANA MEMBUAT DAN MENGGUNAKAN ADT SINGLE LINKED LIST** yang berisi data integer.

Urutan langkahnya adalah sebagai berikut :

1. Membuat deklarasi tipe List

Syntax c++ dapat dilihat di modul halaman 37 atau dapat dilihat pada gambar dibawah ini.

- Pada list.h

```
#include <iostream>
#define first(L) L.first
#define next(P) P->next
#define info(P) P->info
using namespace std;
typedef int infotype;
typedef struct elmlist *address;

struct elmlist {
    infotype info;
    address next;
};
struct List{
    address first;
};
```

- Pada list.cpp

```
#include <iostream>
#include "list.h"
using namespace std;
```

2. Tahap kedua kita akan membuat list kosong, yaitu procedure createlist.

- Tambahkan pada list.h pimitif dari procedure createlist

```
void createlist(List &L);
```

- Tambahkan pada list.cpp implementasi dari procedure createList, sintak c++ sbb:

```
void createList(List &L){
    /** this procedure will initialize
        the list L*/
    first(L) = NULL;
}
```

3. Setelah list sudah ada, selanjutnya kita perlu membuat elemen dengan menggunakan fungsi allocate.

- Tambahkan pada list.h primitif dari fungsi allocate

```
address allocate(intotype x);
```

- Tambahkan pada list.cpp implementasi dari fungsi allocate, sintak c++ sbb:

```
address allocate(intotype x){
    address p = new elmList;
    info(p) = x;
    next(p) = NULL;

    return p;
}
```

4. Setelah List dan elemen sudah ada, maka selanjutnya elemen tersebut harus diinsert ke List agar bisa menjadi elemen list. Proses insert dapat menggunakan procedure Insert First, procedure Insert Last, atau procedure insert After. Pada Tugas Pendahuluan kali ini, akan dicontohkan menggunakan insert first.

- Tambahkan pada list.h primitif procedure insertFirst

```
void insertFirst(List &L, address P);
```

- Tambahkan pada list.cpp implementasi dari proc insertFirst, sintak C++ sbb :
(Note : kali ini tidak ada pengecekan apakah list kosong atau tidak)

```
void insertFirst(List &L, address P){
    /** TODO: Insert the new element pointed
        by P to the first of list L*/
    // YOUR CODES HERE
    //-----
    next(P) = first(L);
    first(L) = P;
    //-----
}
```

5. Setelah proses insert elemen, maka agar bisa mengetahui apakah elemen berhasil diinsertkan, maka kita perlu menampilkan isi list.

- Tambahkan pada list.h primitif procedure printInfo

```
void printInfo (List L);
```

- Tambahkan pada list.cpp implementasi dari proc printInfo, sintak C++ sbb :

```
void printInfo(List L){
    /** this procedure will output
        the info of each element
        in list L*/

    address p = first(L);
    while (p != NULL){
        cout << info(p) << ", ";
        p = next(p);
    }
    cout << endl;
}
```

Selamat Anda sudah selesai membuat ADT Linked List sederhana

6. Sekarang, setelah ADT List sudah terisi dengan beberapa fungsi Procedur di atas, maka mari buat sebuah List berisi 3 elemen yang berisi 3 digit nim terkahir Anda di main.cpp

Adapun gambaran isi dari main.cpp nya adalah sbb :

```
1  #include <iostream>
2  #include "list.h"
3  using namespace std;
4
5
6  int main()
7  {
8      // 1. Panggilah create list
9
10     // 2. Buat sintak menanyakan angka pertama yang ingin dimasukkan user ke list
11
12     // 3. Panggil fungsi allocate agar data tersebut diasikan elemen
13
14     // 4. Panggil procedure insert first yang telah dibuat
15
16     // 5. Panggil procedure show info untuk mengecek apakah angka tersebut berhasil menjadi elemen di List.
17
18     // 6. buat kembali sintak no 2 s/d no 5 untuk data angka kedua dari user
19
20     // 7. buat kembali sintak no 2 s/d no 5 untuk data angka ketiga dari user
21
22
23     return 0;
24 }
25
26
```

Tugas rekan-rekan adalah mengisi main.cpp di atas dengan sintak C++ sesuai dengan petunjuk. Buka kembali modul sebelumnya ya ^_^. Setelah selesai, compile dan run. Lalu **screen capture hasilnya**.

Jika dilayar muncul 3 digit nim Anda yang muncul secara terurut TERBALIK dari urutan input,

SELAMAT ANDA BERHASIL MEMBUAT SINGLE LINKED LIST.

7. Selanjutnya mari tambah procedure deleteFirst

- Tambahkan pada list.h primitif procDelete first

```
void deleteFirst(List &L, address &P);
```

- Tambahkan pada list.cpp implementasi proc deleteFirst

```
void deleteFirst(List &L, address &P){
    /** TODO: Delete the first element
        of list L and assign the
        deleted element to P*/
    // YOUR CODES HERE
    //-----
    if (first(L) != NULL) {
        P = first(L);
        first(L) = next(P);

        next(P) = NULL;
    }
}
```

- Tambahkan pada main.cpp
 - sintak yang memanggil procedure deleteFirst.
 - Sintak yang memanggil printInfo.

Screen Capture hasil running program Anda

Jika hasil run di layar Anda angka pertama dari List hilang maka

SELAMAT ANDA BERHASIL MENDELETE ELEMEN PERTAMA DARI SEBUAH LIST

8. SESI HAVE FUN. Rekan-rekan dapat mencoba hal di bawah ini agar memudahkan saat praktikum:

- Tambahkan procedure insertLast, insertAfter, deleteLast, deleteAfter pada list.h dan list.cpp
- Tambahkan Function searchInfo pada list.h dan list.cpp
- Ubah main.cpp agar proses insert N data tidak satu persatu, tapi sesuai dengan jumlah digit NIM yaitu 10 data (clue : gunakan looping). Dan NIM yang diinput, saat di show tidak boleh terurut terbalik (clue : gunakan insert Last)

Tampilan (underscore adalah inputan user):

```
Masukkan NIM perdigit
Digit 1 : 1
Digit 2 : 1
Digit 3 : 3
Digit 4 : 1
Digit 5 : 9
Digit 6 : 6
Digit 7 : 4
Digit 8 : 7
Digit 9 : 4
Digit 10 : 2
Isi list : 1131964742
```