

JURNAL PRAKTIKUM MODUL 4  
STRUKTUR DATA 2021/2022  
"Single Linked List (Bagian 1)"

**A. Terbimbing (60 Poin)**

1. Buatlah Code::Block Console C++ Project baru.
2. Buatlah file header baru (\*.h), nama file: "mylist.h"
3. Buatlah file implementasi baru (\*.cpp), nama file: "mylist.cpp"

**Spesifikasi / Silahkan tulis ulang dalam Bahasa C++ pada mylist.h (5 poin)**

```
Type mahasiswa <
    nama : string
    nim  : string
    ipk   : float
>

Type infotype : mahasiswa

Type address : pointer to elmList

Type elmList <
    info : infotype
    next : address
>

Type List <
    First : address
>

Procedure createList (input/output L : List)
Function newMahasiswa(nama : string, nim : string, ipk : float) -> infotype
Function newElement (dataBaru: infotype) -> address
Procedure insertFirst (input/output L : List, input P : address)
Procedure insertLast (input/output L : List, input P : address)
Procedure deleteFirst (input/output L : List, output P : address)
Procedure deleteLast (input/output L : List, output P : address)
Procedure printList (input L : List)
```

**Implementasi / Silahkan tulis ulang dalam Bahasa C++ pada mylist.cpp**

**4. (5 poin)**

```
Procedure createList (input/output L : List)
{
    IS. –
    FS. Terbentuk sebuah list di mana, first dan last dari L bernilai NIL. }

Kamus
Algoritma
    First (L) <- NIL
```

## 5. (5 poin)

**Function** newMahasiswa (nama : string, nim : string, ipk : float) -> infotype

{ Fungsi yang membuat variabel baru bertipe data infotype }

**Kamus**

mhs : infotype

**Algoritma**

mhs.nama = nama

mhs.nim = nim

mhs.ipk = ipk

-&gt; mhs

## 6. (5 poin)

**Function** newElement (dataBaru: infotype) -> address

{ Return alamat alokasi memori sebuah elmList yang berisi dataBaru. }

**Kamus**

P: address

**Algoritma**

alokasi(P)

info (P) &lt;- dataBaru

next (P) &lt;- NIL

-&gt; P

## 7. (5 poin)

**Procedure** insertFirst (input/output L : List, input P : address)

{ IS. Terdefinisi pointer P berisi alamat elmList, dan sebuah list L (L mungkin kosong).

FS. elmList yang ditunjuk oleh P ditambahkan ke dalam list sebagai elemen pertama.

}

**Kamus****Algoritma**

if First (L) = NIL then

First (L) &lt;- P

else

next (P) &lt;- First (L)

First (L) &lt;- P

## 8. (5 poin)

**Procedure** deleteFirst (input/output L : List, output P : address)

{ IS. Terdefinisi sebuah list L (L tidak kosong dan mungkin berisi satu elemen).

FS. P berisi alamat elmList yang pertama, elmList yang ditunjuk oleh P dihapus dari list }

**Kamus****Algoritma**

P &lt;- First (L)

if next (First (L)) = NIL then

First (L) &lt;- NIL

else

```
First (L) <- next (First (L))  
next (P) <- NIL
```

## 9. (5 poin)

```
Procedure printList (input L : list);  
{ IS. Terdefinisi sebuah list L  
  FS. Menampilkan semua info elmList di list. }  
Kamus  
P: address  
i : integer  
Algoritma  
  if First(L) = NIL then  
    output("List kosong!")  
  else  
    i <- 1  
    P <- First (L)  
    while P ≠ NIL do  
      output(i)  
      output ("Nama: ", info(P).nama)  
      output ("NIM  : ", info(P).nim)  
      output ("IPK   : ", info(P).ipk)  
      P <- next (P)  
  
    output("List selesai ditampilkan!")
```

## 10. (10 poin)

```
Procedure insertLast (input/output L : List, input P : address)  
{ IS. Terdefinisi pointer P berisi alamat elmList, dan sebuah list L (L tidak kosong).  
  FS. elmList yang ditunjuk oleh P ditambahkan ke dalam list sebagai elemen terakhir. }
```

## 11. (10 poin)

```
Procedure deleteLast (input/output L : List, output P : address)  
{ IS. Terdefinisi sebuah list L (L tidak kosong dan berisi lebih dari satu elemen).  
  FS. P berisi alamat elmList yang terakhir, elmList yang ditunjuk oleh P dihapus dari list }
```

Jalankan program pada main.cpp untuk memastikan function/prosedur sudah benar

**12. main.cpp (5 point)****Kamus**

L: List

P, temp: address

infotype: Mhs

**Algoritma**

createList(L);

printList(L);

Mhs = newMahasiswa ("Alice", "1301190202", 3.5)

P = newElement (Mhs)

insertFirst(L, P);

Mhs = newMahasiswa ("Bob", "1301190203", 4)

P = newElement (Mhs)

insertFirst(L, P);

printList(L);

Mhs = newMahasiswa ("Chihaya", "1301190204", 3.6)

P = newElement (Mhs)

insertLast(L, P)

Mhs = newMahasiswa ("Delta", "1301190205", 2.7)

P = newElement (Mhs)

insertLast(L, P)

Mhs = newMahasiswa ("Euniche", "1301190201", 3.9)

P = newElement (Mhs)

insertFirst(L, P)

printList(L);

deleteFirst(L, temp);

deleteLast(L, temp);

printList(L);

Jika sudah benar, tampilan pada terminal akan sebagai berikut.

```
List kosong!

[1]
Nama: Bob
NIM : 1301190203
IPK : 4
[2]
Nama: Alice
NIM : 1301190202
IPK : 3.5
List selesai ditampilkan!

[1]
Nama: Euniche
NIM : 1301190201
IPK : 3.9
[2]
Nama: Bob
NIM : 1301190203
IPK : 4
[3]
Nama: Alice
NIM : 1301190202
IPK : 3.5
[4]
Nama: Chihaya
NIM : 1301190204
IPK : 3.6
[5]
Nama: Delta
NIM : 1301190205
IPK : 2.7
List selesai ditampilkan!

[1]
Nama: Bob
NIM : 1301190203
IPK : 4
[2]
Nama: Alice
NIM : 1301190202
IPK : 3.5
[3]
Nama: Chihaya
NIM : 1301190204
IPK : 3.6
List selesai ditampilkan!
```

## B. Mandiri (30 Poin)

Sebelum anda mengerjakan Tugas Mandiri yang ada di bawah, lakukan komen untuk setiap baris yang ada di dalam fungsi main pada file main.cpp

### 1. (15 Poin) Implementasikanlah instruksi berikut ke dalam main.cpp!

Program menerima input mahasiswa, dengan ketentuan sebagai berikut:

- Bila NIM anda dimodulo dengan 3 hasilnya 0, maka Input akan berhenti ketika atribut nama yang diinputkan bernilai "-".
- Bila NIM anda dimodulo dengan 3 hasilnya 1, maka Input akan berhenti ketika atribut NIM yang diinputkan bernilai "-".
- Bila NIM anda dimodulo dengan 3 hasilnya 2, maka Input berhenti ketika atribut nama dan NIM yang diinputkan bernilai "-".

Untuk setiap penginputan elemen mahasiswa, insert ke dalam list menggunakan `insertFirst` untuk anda yang memiliki NIM genap, dan `insertLast` untuk anda yang memiliki NIM ganjil.

### 2. (15 Poin) Setelah nomor 1 selesai, (Implementasikan dalam main.cpp juga) lakukan hapus data hingga list kosong, dan setiap kali sebuah elemen dihapus, outputkan elemen tersebut (dengan menampilkan ketiga atribut pada elemen mahasiswa tersebut).

Ketentuan hapus data sebagai berikut:

- Bila NIM anda genap, lakukan delete first pada list
- Bila NIM anda ganjil, lakukan delete last pada list

Algoritma penghapusan data berhenti ketika list kosong. Kemudian panggil prosedur `printList(L)` untuk memastikan bahwa list sudah kosong.

~Selamat Mengerjakan^^