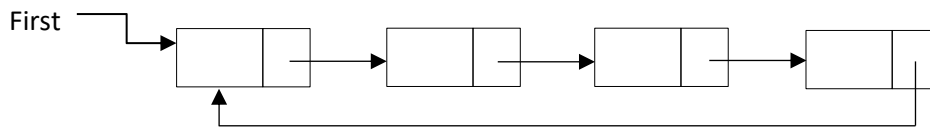


# JURNAL PRAKTIKUM MODUL 5

## MK STRUKTUR DATA 2019/2020-2

### Cicular Single Linked-List dengan Pointer First

Sebuah Circular Single Linked List dengan pointer kepala First digunakan untuk menyimpan data berupa informasi proses/aplikasi yang berjalan pada suatu sistem operasi. Karena biasanya user menjalankan banyak aplikasi bersamaan di PC, maka perlu diatur sedemikian sehingga ada mekanisme untuk menyimpan beberapa aplikasi/proses tersebut untuk selanjutnya dieksekusi oleh CPU secara bergantian. Data yang simpan yaitu nama proses (string), prioritas (int) dan sisa durasi (int). Berikut ilustrasinya:



Buatlah ADT nya (SLL\_Circular.h, SLL\_Circular.cpp, dan Test\_SLL\_Circular.cpp)!

#### A. LATIHAN TERBIMBING

##### 1. Spesifikasi (Silakan ditulis ulang dalam Bahasa C++)

```
Type infotype <
    nama : string
    prioritas : integer
    sisa_durasi : integer
>
Type address : pointer to elmList
Type elmList <
    info : infotype
    next : address
>
Type List <
    First : address
>

Procedure createList(input/output L : List)
Function createElemen(dataBaru: infotype) → address
Procedure insertFirst(input/output L : List, input P : address)
Procedure insertLast(input/output L : List, input P : address)
Procedure InsertAfter(input Prec : address, P : address);
Procedure deleteFirst(input/output L : List, output P : address)
Procedure deleteLast(input/output L : List, output P : address)
Procedure deleteAfter(input Prec: address, output P : address)
Procedure printList(input L : List);
function panjangList(input L : List) → integer
function cariElm(L:list, namaProses : string) → address
```

##### 2. Implementasi (Silakan ditulis ulang dalam Bahasa C++)

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

**Function** createElemen (dataBaru: infotype) → address

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

**Procedure** insertFirst (input/ouput 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. }

**Procedure** insertLast (input/ouput 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 terakhir. }

**Procedure** InsertAfter (input Prec : address, P : address);

{IS. Terdefinisi pointer Prec dan P berisi alamat elmList. Prec mungkin NIL

FS. elmList yang ditunjuk oleh P ditambahkan ke dalam list setelah elmList yang ditunjuk oleh Prec.}

**Procedure** deleteFirst (input/ouput 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}

**Procedure** deleteLast (input/ouput 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}

**Procedure** deleteAfter (input Prec: address, output P : address)

{IS. Terdefinisi pointer Prec berisi alamat elmList, Prec mungkin NIL

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

**Procedure** printList (input L : list);

{IS. Terdefinisi sebuah list L

FS. Menampilkan semua info elmList di list. }

**Function** panjangList(input L : List) → integer

{IS. Terdefinisi sebuah list L (mungkin kosong)

FS. Dikembalikan jumlah elemen di dalam list L}

**function** cariElm(L:list, X : string) → address

{ Return alamat memori elemen list yang memiliki info nama proses == nama, atau NIL apabila tidak ditemukan}

### 3. Program Test\_DLL\_Circular (Silakan ditulis ulang dalam Bahasa C++)

#### Kamus

L: List

P,Q : address

proses : infotype

#### Algoritma

createList (L)

printList (L)

process.nama ← "word"; process.prioritas ← 5 ; process.sisa\_durasi ← 10

P ← createElemen(process)

insertLast (L, P)

printList (L)

process.nama ← "excel"; process.prioritas ← 4 ; process.sisa\_durasi ← 20

P ← createElemen(process)

# JURNAL PRAKTIKUM MODUL 5

## MK STRUKTUR DATA 2019/2020-2

### Cicular Single Linked-List dengan Pointer First

```
insertLast (L, P)
printList (L)
process.nama ← "dota"; process.prioritas ← 100 ; process.sisa_durasi ← 30
P ← createElemen(process)
insertFirst(L, P)
printList(L)
deleteLast (L, P)
printList (L)
deleteFirst (L, P)
printList (L)
deleteFirst (L, P)
deleteFirst (L, P)
deleteFirst (L, P)
printList (L)
Q ← cariElm(L,"dota")
process.nama ← "sinchan"; process.prioritas ← 50 ; process.sisa_durasi ← 30
P ← createElemen(process)
insertAfter(Q,P)
printList (L)
deleteAfter(Q,P)
printList (L)
process.nama ← "sinchan"; process.prioritas ← 50 ; process.sisa_durasi ← 30
P ← createElemen(process)
insertFirst(L, P)
printList(L)
```

Ilustrasi program setelah dijalankan.

```
<word, 5, 10>
<word, 5, 10>, <excel, 4, 20>
<dota, 100, 30>, <word, 5, 10>, <excel, 4, 20>
<dota, 100, 30>, <word, 5, 10>
<word, 5, 10>
```

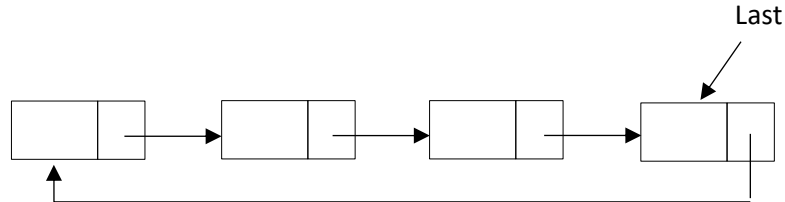
```
<sinchan, 50, 30>
```

### Cicular Single Linked-List dengan Pointer First

#### B. TUGAS TERBIMBING

Modifikasi struktur list menjadi seperti ini (List < Last : Address >). Lalu modifikasi Procedure insertFirst dan insertLast. Jelaskan apa perbedaannya implementasinya!

Hint : First bisa didapatkan dari next(Last(L)).



**Procedure** insertFirst (input/ouput 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**

**Procedure** insertLast (input/ouput 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 terakhir. }

**Kamus**

**Algoritma**

Jelaskan apa perbedaannya yang paling terlihat pada implementasi insertFirst dan insertLast antara menggunakan pointer First dan menggunakan pointer Last dalam struktur List!