

# Jurnal Modul 6

## STRUKTUR DATA – Ganjil 2024/2025

### " DLL "

#### A. Soal Jurnal PART 1

Sebuah Double Linked List dengan pointer kepala First dan Last digunakan untuk menyimpan data browser history. Berikut ilustrasinya:



Buatlah ADT nya (DLL.h, DLL.cpp, dan Main.cpp)!

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

```
Type history <
    title: string
    url: string
    timestamp: string
    visitcount: int
>

Type infotype : history
Type address : pointer to elmList
Type elmList <
    info : infotype
    next : address
    prev : address
>

Type List <
    First : address
    Last : address
>
```

```
Procedure createList (input/output L : List)
Function createElemen (dataBaru: infotype) → address
Procedure insertLast (input/output L : List, input P : address)
Procedure InsertAfter (input/output 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/output Prec: address, output P :
address)
```

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

**Procedure** createList (**input/ouput** L : List)

{IS. –

FS. Terbentuk sebuah list di mana, first dan last dari L bernilai NIL. }

**Kamus**

**Algoritma**

First (L)  $\leftarrow$  NIL

Last (L)  $\leftarrow$  NIL

**Function** createElemen (dataBaru: infotype)  $\rightarrow$  address

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

**Kamus**

P: address

**Algoritma**

alokasi (P)

info (P)  $\leftarrow$  dataBaru

next (P)  $\leftarrow$  NIL

prev (P)  $\leftarrow$  NIL

return P

**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**

if First (L) = NIL then

First (L)  $\leftarrow$  P

Last (L)  $\leftarrow$  P

else

next (Last(L))  $\leftarrow$  P

prev (P)  $\leftarrow$  Last(L)

Last (L)  $\leftarrow$  P

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

{IS. Terdefinisi pointer Prec dan P berisi alamat elmList. Prec  $\neq$  Last(L).

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

**Kamus**

**Algoritma**

next (P)  $\leftarrow$  next (Prec)

```
prev(next(Prec))  $\leftarrow$  P  
next (Prec)  $\leftarrow$  P  
prev (P)  $\leftarrow$  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 }*

**Kamus**

**Algoritma**

```
P  $\leftarrow$  First (L)  
if next (First (L)) = NIL then  
    First (L)  $\leftarrow$  NIL  
    Last (L)  $\leftarrow$  NIL  
else  
    First (L)  $\leftarrow$  next (First (L))  
    prev (First(L))  $\leftarrow$  NIL  
    next (P)  $\leftarrow$  NIL
```

**//DEFINISIKAN PROCEDURE DELETELAST!**

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

*{IS. Terdefinisi pointer Prec berisi alamat elmList. Prec  $\neq$  Last(L). next(Prec)  $\neq$  Last(L). FS.*

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

**Kamus**

**Algoritma**

```
P  $\leftarrow$  next (Prec)  
next (Prec)  $\leftarrow$  next (P)  
prev (next(P))  $\leftarrow$  prec  
next (P)  $\leftarrow$  NIL  
prev (P)  $\leftarrow$  NIL
```

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

*{IS. Terdefinisi sebuah list L }*

*FS. Menampilkan semua info elmList di list. }*

**Kamus**

**Algoritma**

```
P  $\leftarrow$  First (L)  
while P $\neq$ NIL do  
    output (info (P))  
    P  $\leftarrow$  next (P)
```

### 3. Program Utama (Silahkan ditulis ulang dalam Bahasa C++)

#### Kamus

L: List

H: infotype

Elm: address

#### Algoritma

createList (L)

printList (L)

input (H.title, H.url, H.timestamp, H.visitcount)  
{inputkan "example", "https://www.exsample.com",  
"2024-10-18/10:00:00", 1}

Elm  $\leftarrow$  createElemen (H)

insertLast (L, Elm)

printList (L)

input (H.title, H.url, H.timestamp, H.visitcount)  
{inputkan "OpenAI", "https://www.openai.com",  
"2024-10-18/10:15:00", 9}

Elm  $\leftarrow$  createElemen (H)

insertLast (L, Elm)

printList (L)

input (H.title, H.url, H.timestamp, H.visitcount)  
{inputkan "GitHub", "https://www.github.com",  
"2024-10-18/10:19:53", 17}

Elm  $\leftarrow$  createElemen (H)

insertLast (L, Elm)

printList (L)

deleteFirst(L, Elm)

printList(L)

deleteLast(L, Elm)

printList(L)

deleteLast(L, Elm)

printList(L)

SOAL JURNAL PART 2

1. Buat implementasi procedure berikut ini (Implementasikan juga pada header):

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

**Procedure** insertAscending (input/ouput L : List, input dataBaru : infotype)  
{IS. Terdefinisi sebuah data, dan sebuah list L (L mungkin kosong, jika list sudah ada isinya, asumsi isi list sudah terurut menaik berdasarkan "visitcount").  
FS. dataBaru ditambahkan ke dalam list dengan aturan: data di dalam list harus selalu terurut secara menaik (ascending) berdasarkan "visitcount".  
Note: Gunakan procedure insertFirst, insertLast, dan insertAfter yang sudah dibuat sebelumnya. }

**Function** findElemen (L: List, X: string) → boolean  
{Mencari sebuah elemen dengan title X dalam list L. Kembalikan "true" jika ditemukan dan "false" jika tidak ditemukan.

Ilustrasi:



Elemen dengan title "OpenAI" ditemukan di List.  
}

**Function** moreThanOnce (L: List) → integer  
{IS. List mungkin kosong.  
FS. Mengembalikan jumlah History yang memiliki visitcount lebih dari 1. Jika list kosong atau elemen tidak ditemukan di list, maka kembalikan 0.

Ilustrasi:



Elemen yang memiliki visitcount lebih dari 1 ditemukan di list (true).  
Elemen yang memiliki visitcount lebih dari 1 muncul 2 kali pada list  
}

2. Buat program utama seperti berikut:

### Kamus

L: List

H: infotype

Elm: address

### Algoritma

createList (L)

printList (L)

input (H.title, H.url, H.timestamp, H.visitcount)  
{inputkan "example", "https://www.exsample.com",  
"2024-10-18/10:00:00", 1}

Elm  $\leftarrow$  createElemen (H)

insertAscending (L, Elm)

printList (L)

input (H.title, H.url, H.timestamp, H.visitcount)  
{inputkan "GitHub", "https://www.github.com",  
"2024-10-18/10:19:53", 17}

Elm  $\leftarrow$  createElemen (H)

insertAscending (L, Elm)

printList (L)

input (H.title, H.url, H.timestamp, H.visitcount)  
{inputkan "OpenAI", "https://www.openai.com",  
"2024-10-18/10:15:00", 9}

Elm  $\leftarrow$  createElemen (H)

insertAscending (L, Elm)

printList (L)

output(findElemenElemen(L, "OpenAI"))

output(moreThanOnce(L))